Doctoral Programme in Mathematical Models and Methods in Engineering - XXXIV cycle

Efficient reduced order modeling for nonlinear parameterized problems in elastodynamics: application to cardiac mechanics

Doctoral dissertation of Ludovica Cicci

POLITECNICO
MILANO 1863

Advisor: Prof. Alfio Quarteroni

Co-advisor: Prof. Andrea Manzoni

Chair of the doctoral program: Prof. Irene Sabadini

Abstract

In this Thesis we propose new computationally efficient solutions to nonlinear, parameterized time-dependent problems by means of reduced order models (ROMs). More specifically, we propose fast and accurate approximations of elastodynamics problems arising in cardiac mechanics. Being able to perform efficient numerical simulations in this context is indeed essential to explore multiple virtual scenarios, to quantify cardiac outputs and related uncertainties, as well as to evaluate the impact of pathological conditions. All these tasks call for repeated model evaluations over different input parameter values, thus making usual high-fidelity, full order models, such as those based on the finite element method, computationally prohibitive. Alternative numerical methods have been developed aiming to compute reliable solutions to parametric partial differential equations (PDEs) at a greatly reduced computational cost. Among these, the reduced basis (RB) method represents a powerful and widely used technique, characterized by a Galerkin projection of the problem onto a low-dimensional subspace and by a splitting of the reduction procedure into a costly offline phase and an inexpensive online phase. However, in the case of nonlinear problems, the online stage has a cost that still depends on the high-fidelity dimension, as the assembling of the reduced operators requires the reconstruction of the high-fidelity ones. In this Thesis we exploit the RB method, equipped with hyper-reduction techniques for handling the nonlinear terms, for the solution to time-dependent problems arising in cardiac mechanics, where the complex material behavior (the cardiac tissue) is described by means of an exponentially nonlinear constitutive law accounting for the presence of myocardial fibers. Some examples related to the use of hyper-ROMs in the multi-query contexts of sensitivity analysis and parameter estimation are also given. Nonetheless, severe challenges arise from the approximation of the nonlinear terms, usually requiring a large number of basis functions to correctly capture their variability, thus compromising the overall online efficiency of the ROM. To overcome the computational bottleneck of hyper-ROMs, we develop a novel projection-based, deep learning-based, ROM, that we name Deep-HyROMnet. The key idea is to combine the Galerkin-RB approach with deep neural networks (DNNs) to approximate reduced nonlinear operators efficiently. Unlike data-driven strategies, for which the predicted output is not guaranteed to satisfy the underlying PDE, Deep-HyROMnet is a physics-based ROM, as it computes the problem solution by solving a reduced nonlinear system. A further benefit of the proposed method lies on the fact that the inputs given to the DNNs are low-dimensional arrays, so that overwhelming training times and costs can be avoided. We show how our model outperforms classical hyper-ROMs (such as POD-Galerkin-DEIM ROMs exploiting the discrete empirical interpolation method) in terms of computational speed-up for the solution of a wide range of problems in nonlinear elastodynamics, still achieving accurate results. Finally, we demonstrate the performances of Deep-HyROMnet on patient-specific cardiac geometries involving about 127000 structural degrees of freedom, and consider the case of a 3D mechanics model that is monolithically coupled to a 0D windkessel model for blood circulation, to simulate the cardiac functions in both physiological and pathological scenarios.

Contents

Introduction

Mathematical models involving partial differential equations (PDEs) are ubiquitous in applied sciences and engineering, being able to describe a wide variety of phenomena, including the complex mechanisms of cardiac physiology. Computer simulations reproducing in-silico heart functions, both in physiological and in pathological scenarios, allow to investigate the impact of various clinical conditions on the cardiac cycle, to explore potentially different surgical procedures and to measure non-invasively clinical quantities of interest, thus providing a great opportunity to improve diagnosis, provide real-time decision support and predict prognosis [QMV17]. However, numerical simulations are usually hampered by severe computational costs, which prevent their use in clinical practice.

Several models have been developed in the last decades to describe single or coupled cardiac functions [Tra11; Nor+11; Cha+16; Hir+17; Qua+19; Zin+21], such as tissue mechanics, blood dynamics and electrophysiology. The work of this Thesis is inspired by the mechanics problem, which is governed by the equation of motion in elastodynamics, thus relying on finite elasticity theory to describe the heart contraction and relaxation. Modeling the mechanics of the cardiac tissue requires complex constitutive laws, characterized by an exponential strain energy function and by the presence of muscular fibers and sheets. For this reason, the passive myocardium is considered as an hyperelastic orthotropic material [GCM95], ultimately yielding nonlinear mechanics problems to be solved. Furthermore, the heart muscle contracts after being electrically activated without the need of an external load, so that the active behavior of the tissue has to be properly taken into account.

Indeed, cardiac models depend on a large set of input parameters that are defined to characterize, e.g., material properties, loads, boundary/initial conditions, source terms, or geometrical features, and which influence the solution to the problem under investigation and the associated outputs of interest, i.e. suitable quantities than depend on the problem solution itself. The knowledge of these input values is often severely limited due to both the difficulties in performing experiments and the uncertainty associated with their measurements; moreover, some parameters are related to the mathematical model and do not have a direct physical meaning, while others are hampered by intrinsic variability among patients. In this context characterizing the impact of parameter variation on quantities of clinical interest [LF+20; Cam+20; RCSR19], understanding how uncertainty in the model inputs affects the considered outputs [Mir+16; Hau+18] and estimating the input values from available data [Cha+13; Hir18] play a key role in providing reliable patient-specific simulations. From a numerical perspective, these tasks yield the solution of sensitivity analysis, uncertainty quantification and parameter estimation problems. All these scenarios share the need of calling for multiple queries of the parameter-to-solution map, or at least input-output map, in accurate (and possibly very fast) way.

Despite the continuous increase of available computing resources and the development of new numerical methods, high-fidelity simulations based on classical discretization schemes, like the finite element method, entails huge computational costs in terms of both CPU time and memory, as a large number of degrees of freedom is usually required. Computational complexity is even more exacerbated if one is interested in going beyond a single, direct simulation, so that relying on high-fidelity full order model (FOMs) for the approximation of the underlying PDE solution may become computationally unfeasible. To address this issue, alternative techniques, such as lower-fidelity models, statistical emulators or reduced order models (ROMs), have been developed over the years, aiming to compute reliable solutions at a greatly reduced computational cost.

Objective and contributions of this work

Motivated by the aforementioned needs, the main goal of this Thesis is to develop accurate projection-based ROMs that enable the efficient solution to nonlinear, time-dependent parameterized PDEs by seeking an approximation of the unknown state solution by performing a Galerkin projection of the discretized governing equations onto a suitable low-dimensional space [BGW15; Ben+17; QMN16]. We focus in particular on cardiac mechanics, however considering simplified problems and preliminary geometries, with the aim of providing a powerful tool to address the solution to multi-query problems and to perform (almost) real-time numerical simulations. Exploiting proper orthogonal decomposition (POD) for the construction of the reduced subspace, we can rely on completely physics-based ROMs, which can be extremely convenient rather than using fully data-driven emulators or surrogate models providing inexpensive evaluations of the input-output queries. In fact, by retaining the underlying structure of the physical model and providing an approximation at each time step of the displacement vector, physics-based ROMs provide approximations that are physically consistent and reliable, thanks to the approximability properties of the low-dimensional spaces provided by POD. Not only, physics-based ROMs allow to consider new quantities of interest that might be useful both in uncertainty quantification studies or in medical research without the need to repeat the whole reduction strategy or training process.

In particular, we consider POD-Galerkin ROMs built through the reduced basis (RB) method [QMN16; HRS16], which represents a powerful and widely used technique characterized by a splitting of the reduction procedure into an expensive, parameter-independent offline phase and a usually inexpensive, parameter-dependent online phase. The efficiency of the RB method mainly relies on two crucial assumptions, namely that

- the solution manifold is low-dimensional, so that the FOM solutions can be approximated as a linear combination of few reduced modes with a small error;
- the online stage is completely independent of the high-fidelity dimension [Far+20].

The first assumption concerns the approximability of the solution manifold, that is the set of PDE solutions for all input values varying in the parameters space, and is associated with the slow decay of the Kolmogorov N-width [Pin85]. Despite their nonlinear nature, parameterized problems in elastodynamics, including those arising from cardiac modeling, do not pose serious issues regarding the dimensionality of the reduced spaces, the slow decay of singular values when considering POD, or the assumption stated by the RB method entailing a linear superimposition

of modes. Nonetheless, the second assumption regarding the offline-online decoupling cannot be fulfilled when dealing with nonlinear problems, as the assembling of the reduced nonlinear operators at each Newton iteration requires the reconstruction of the high-fidelity ones, so that the online stage has a cost that still depends on the FOM dimension.

To avoid this drawback, we need to recover an affine parametric dependence of the nonlinear terms, thus relying on a further level of approximation. Among several hyper-reduction techniques, *project-then-approximate* strategies have been introduced aiming at approximating directly ROM operators. In this Thesis we consider the discrete empirical interpolation method (DEIM) [CS10], which is based on the idea to replace both the reduced residual vectors and the corresponding reduced Jacobian matrices by means of a collateral reduced basis expansion computed through an interpolation procedure onto a (hopefully small) set of points, so that only a subset of entries of the nonlinear arrays needs to be assembled during Newton iterations at the ROM level. So far, only few contributions have considered the construction of projection-based ROMs addressing nonlinear unsteady problems related with cardiac mechanics [BMQ17; MBQ18; Cha+13; Pfa+20]. Numerical experiments showed that, despite their highly nonlinear nature, elastodynamics problems can be reduced by exploiting projection-based strategies in an effective way, with POD-Galerkin ROMs achieving very good accuracy even in presence of a handful of basis functions. In this respect, mechanics problems do not pose the same issues as transport, wave, or convection-dominated phenomena, where the RB method may yield inefficient, high-dimensional ROMs. Furthermore, we show how POD-Galerkin-DEIM ROMs can be used to alleviate the computational costs of global sensitivities analysis and parameter estimation problems carried out by means of a standard sampling techniques, leading to comparable results with those obtained using expensive FOMs.

Indeed, if the construction of a reduced subspace to approximate the problem solution does not pose serious issues, resulting in extremely low-dimensional spaces, numerical results highlight that the computational bottleneck is represented by the assembling of DEIM reduced operators and their projection. As a matter of fact, a large number of interpolation points are required to ensure the convergence of the reduced Newton systems arising from the linearization of the nonlinear hyper-reduced ROM when dealing with highly nonlinear elastodynamics problems with complex, anisotropic material laws. In order to overcome the limitations associated with POD-Galerkin-DEIM, we propose a new semi-intrusive, deep learning-enhanced model order reduction strategy, which hereon we refer to as Deep-HyROMnet. Since the efficiency of nonlinear POD-Galerkin ROMs hinges upon the cost-effective approximation of the projections of the (discrete) nonlinear operators, the key idea is to leverage a Galerkin-RB method for the solution-space reduction and to rely on deep neural networks (DNNs) to perform hyper-reduction at a greatly reduced cost. Unlike data-driven-based methods, for which the predicted output is not guaranteed to satisfy the underlying PDE, Deep-HyROMnet is still a physics-based model, as it computes the ROM approximation of the displacement field by solving the reduced nonlinear systems using Newtons method. By dramatically reducing the CPU time required for the approximation of the reduced nonlinear operators, we are able to compute the solution to the 3D cardiac mechanics model monolithically coupled to a 0D windkessel model for blood circulation in a bunch of seconds on a patient-specific left ventricle geometry, even when fine computational meshes are required, making a step forward the integration of computational models into the medical practice.

Outline of the Thesis

This Thesis is organized as follows.

Chapter 1
: We introduce the essential features of continuum mechanics and finite deformation theory, which provide the fundamental framework for the modeling of elastodynamics problems. Then, after an overview of the cardiac functions, we focus on the mathematical models for tissue mechanics used for the description of the passive and the active behavior of the cardiac muscle. Finally, we derive the fully discrete FOM arising from the high-fidelity approximation of the nonlinear, parameterized initial boundary-value problem for the elastodynamics equation by means of the finite element method and implicit time integration schemes.

Chapter 2
: In order to provide the necessary tools to understand the reduction strategies adopted in this Thesis, we briefly described the Galerkin-RB method in the case of stationary PDEs featuring an affine parametric dependence. Afterwards, we show how to perform the construction of POD-Galerkin-DEIM hyper-reduced models, relying on

  i POD or randomized singular value decomposition for finding a low-dimensional trial subspace;

  ii Galerkin projection to generate, in a physically consistent way, a reduced order model;

  iii hyper-reduction through DEIM technique to accelerate the assembling of nonlinear (with respect to the solution, to input parameters, or to both) terms.

  The numerical performance of POD-Galerkin-DEIM ROMs are investigated on different applications, from structural tests on simple geometries to benchmark problems in cardiac mechanics, simulating different phases of the cardiac cycle on both idealized and realist geometries.

Chapter 3
: We show how to take advantage of POD-Galerkin-DEIM ROMs to efficiently address the solution to multi-query uncertainty quantification tasks. We perform global variance-based sensitivity analysis studies by computing Sobol' indices, thus providing a very useful indicator of which input factors should be correctly chosen and which other make no significant contribution to the variance of the outputs of interest. Identifying the less influential factors on a given output quantity, which can then be fixed at any given value in their range of variability, is extremely useful also in view of the construction of ROMs, as it allows significant model simplifications. Furthermore, we perform Bayesian calibration for the estimation of material parameters in a structural mechanics application by means of Markov Chains Monte Carlo. Relying on POD-Galerkin-DEIM ROMs allows a remarkable reduction of the computational time (from days to hours) with respect to the FOM, still preserving the overall accuracy.

Chapter 4

We present different deep learning-based strategies to efficiently handle the reduced residual vectors and Jacobian matrices, based on both Newton and Broyden methods, and we describe the DNN architecture employed for the computation of the approximated nonlinear reduced operators. Numerical examples show that the Newton approach has to be preferred to the ones based on Broyden method for the construction of the hyper-ROMs. The performances of Deep-HyROMnet are then compared to POD-Galerkin-DEIM ROMs in several scenarios, both in terms of accuracy and efficiency. In particular, we consider a series of structural tests on a rectangular geometry with different loading boundary conditions and two benchmark tests on a prolate spheroid geometry simulating cardiac relaxation and contraction. The results highlights the advantages of avoiding the assembling stage of the nonlinear terms required by standard hyper-reduction techniques by showing a great improvement of the CPU speed-up achieved by Deep-HyROMnet with respect to the FOM for computing the PDE solution for every new instance of the input parameters.

Chapter 5

To conclude, we address the efficient solution to parameterized cardiac mechanics problems by means of Deep-HyROMnet hyper-reduced models. In particular, we introduce a 3D-0D coupled structure-circulation model for the left ventricle, taking into account contractile forces by means of a surrogate active tension. We show how our DNN-based hyper-ROM can be adapted to different test cases in this context and finally present two numerical examples on a patient-specific left ventricle geometry, the former focusing on a physiological scenario and the latter assuming the presence of an ischemic region inside the myocardium, also demonstrating how the developed framework is able to enhance forward uncertainty quantification analysis otherwise unaffordable.

This Thesis has been elaborated within the "iHEART - An Integrated Heart Model for the simulation of the cardiac function" project, focusing on the task of *Variability and Uncertainty* for the development of efficient and reliable RB methods for parameter dependent numerical simulations, as well as computational techniques for both forward and inverse UQ problems. The iHEART project has been funded with an Advanced European Research Council (ERC) grant (agreement No 740132, P.I. Prof. A. Quarteroni) with the aim to create a complete mathematical model of the human heart, which includes all the relevant physiological processes. From the implementation point of view, the reduced order models considered in this Thesis have been implemented from scratch in Python in our software package `pyfe`$^{\texttt{x}}$, which contains a Python binding with the in-house Finite Element library `life`$^{\texttt{x}}$ (`https://lifex.gitlab.io/`), a high-performance C++ library developed within the iHEART project and based on the `deal.II` (`https://www.dealii.org`) Finite Element core [Arn+20]. To perform sensitivity analysis and inverse uncertainty quantification we have taken advance of the Dakota toolkit [Ada+20]. Neural networks required by our Deep-HyROMnet technique have been implemented by means of the Tensorflow deep learning framework [Aba+16]. The training phase has been carried out on a workstation equipped with an Nvidia GeForce GTX 1070 8 GB GPU. Finally, post-processing of the computational results has been performed using MATLAB. All the numerical simulations have been performed on a PC desktop computer with 3.70GHz Intel Core i5-9600K CPU and 16GB RAM.

Chapter 1

The Elastodyamical problem: formulation and high-fidelity approximation

This chapter is intended to introduce the essential features of continuum mechanics and to provide a general framework for the modeling of elastodynamics problems, such as those arising in cardiac mechanics. First of all, we provide the basic notions of nonlinear continuum mechanics. Since biological tissues commonly experience large deformations [Dem81], the central assumption to small strain theory does not hold, and one needs to rely on the more accurate, yet complex, finite deformation theory. The driving application is that of cardiac physiology, for which we will focus on the mathematical description of both active and passive behaviors of the cardiac muscle. Finally, we introduce the high-fidelity model arising from the discretization of the elastodynamics problem, which represents the reference full order model considered in the subsequent reduction strategies.

1.1 Nonlinear computational mechanics

The methods of continuum mechanics represent an effective tool to describe various physical phenomena without taking into account internal micro structures. In this framework the physical object under investigation is treated as a continuous medium, that is, a composition of a continuous set of particles, characterized by suitable field quantities, whose motion is described by means of balance equations (such as the one imposing the equilibrium of forces or momentum), ultimately yielding the governing equations of elastodynamics.

To characterize relevant quantities in continuum mechanics, it is crucial to define a frame of reference a continuum is observed from, for which two different approaches can be usually adopted: the Lagrangian and the Eulerian descriptions. The first pays attention to a body particle with respect to its initial coordinate frame (that is, where the body was before deformation), while the second describes these quantities in terms of the position acquired by the body during deformation, by following the trajectory of any point in space. The Lagrangian description is commonly used in solid mechanics, as continua constitutive behavior is usually given in terms of material coordinates and will be the approach adopted in this Thesis. For the detailed theory we refer, e.g., to [Hol00] and [BW97].

1.1.1 Kinematics

Given a continuum body $\mathcal{B}$ embedded in a three-dimensional Euclidean space, let $\Omega_0 \subset \mathbb{R}^3$ be its reference configuration at time $t = 0$ and $\Omega_t \subset \mathbb{R}^3$ its current configuration at time $t > 0$. The motion of the body $\boldsymbol{\chi} : \Omega_0 \times \mathbb{R}^+ \to \mathbb{R}^3$ is defined as a function which takes a generic

material point $\mathbf{X} \in \Omega_0$ and maps it onto the corresponding spatial point $\mathbf{x} = \boldsymbol{\chi}(\mathbf{X}, t) \in \Omega$, for all times $t > 0$. The displacement field is defined as

$$\mathbf{u}(\mathbf{X}, t) = \mathbf{x}(\mathbf{X}, t) - \mathbf{X},$$

for all times $t \geq 0$, and represents the unknown of the problem we are interested in.

A central quantity in finite deformations theory is the so-called deformation gradient,

$$\mathbf{F}(\mathbf{X}, t) = \nabla_0 \boldsymbol{\chi}(\mathbf{X}, t) = \frac{\partial \mathbf{x}(\mathbf{X}, t)}{\partial \mathbf{X}}, \quad \mathbf{F}_{ij} = \frac{\partial x_i(\mathbf{X}, t)}{\partial X_j},$$

which describes the relationship between quantities in the undeformed and the deformed configurations of the body, and can be expressed in terms of the displacement field as

$$\mathbf{F}(\mathbf{X}, t) = \mathbf{I} + \nabla_0 \mathbf{u}(\mathbf{X}, t),$$

where $\mathbf{I}$ is the identity matrix and ∇_0 denotes the material gradient. The change in volume between the reference and the current configurations at time $t > 0$ is given by the determinant of the deformation tensor,

$$J(\mathbf{X}, t) = \det \mathbf{F}(\mathbf{X}, t),$$

known as volume ratio. If there is no motion, i.e. $\mathbf{x} = \mathbf{X}$, we obtain the consistency condition $J = 1$. In general, a motion for which this condition holds is said to be *isochoric*.

Another widely used measure of deformation is the right Cauchy-Green deformation tensor $\mathbf{C} = \mathbf{F}^T\mathbf{F}$, which operates on material vectors and is invariant with respect to rigid body motion, whilst a common measure of the strain is the Green-Lagrange strain tensor

$$\mathbf{E} = \frac{1}{2}(\mathbf{C} - \mathbf{I}),$$

which is defined in the reference configuration and is zero in absence of motion. All these kinematic quantities are commonly used in the development of constitutive equations describing the relationship between mechanical forces and the material motion.

1.1.2 Stresses and constitutive equations

We now introduce the concept of stress and other important related quantities. We point out that, since stresses are typically defined in the current configuration (as force per unit area) in this section we adopt both the spatial and material descriptions.

Let $\mathbf{t} \in \mathbb{R}^3$ be the Cauchy traction vector, that is, the force measured per unit surface area defined in the current configuration, and $\mathbf{T} \in \mathbb{R}^3$ the (pseudo[1]) surface traction in the reference configuration, such that

$$d\mathbf{f} = \mathbf{t}ds = \mathbf{T}dS.$$

The Cauchy stress theorem states that there exist unique second-order tensor fields $\boldsymbol{\sigma}$ and $\mathbf{P}$ such that

$$\begin{aligned} \mathbf{t}(\mathbf{x}, t, \mathbf{n}) &= \boldsymbol{\sigma}(\mathbf{x}, t)\mathbf{n}, \\ \mathbf{T}(\mathbf{X}, t, \mathbf{N}) &= \mathbf{P}(\mathbf{X}, t)\mathbf{N}, \end{aligned} \tag{1.1}$$

[1]It acts on the current boundary region $\partial\Omega$ and is a function of the material position $\mathbf{X}$ and the outward normal $\mathbf{N}$ to the reference boundary $\partial\Omega_0$.

where $\mathbf{n}$ and $\mathbf{N}$ denote the outward unit normal to the boundary surface $\partial\Omega$ and $\partial\Omega_0$, respectively. Stress can thus be seen as the effects of the internal force that neighboring infinitesimal volume elements mutually exert on each other [ES19].

The following relation

$$\boldsymbol{\sigma} = J\mathbf{P}\mathbf{F}^{-T}$$

holds between the Cauchy stress tensor $\boldsymbol{\sigma}$ and the first Piola-Kirchhoff stress tensor $\mathbf{P}$. Similarly to $\mathbf{F}$, the Piola stress $\mathbf{P}$ is a two-point tensor, which relates a surface area in the reference configuration to the corresponding force vector in the current configuration.

The governing equations of solid mechanics, that will be introduced in the following subsection, are written in terms of the stresses inside the body, which, on their hand, result from the body deformation and must be expressed in terms of some measure of this deformation, such as strain [ES19]. The relationships between stresses and strain are known as *constitutive laws* and characterize the response of the material under consideration. In this Thesis we restrict ourselves to the case of hyperelastic materials, whose behavior only depends on the current deformation state. For this particular class of elastic materials, we can assume that there exists a strain energy density function $\mathcal{W}$: $Lin^+ \to \mathbb{R}$ such that

$$\mathbf{P}(\mathbf{F}) = \frac{\partial \mathcal{W}(\mathbf{F})}{\partial \mathbf{F}}$$

gives the functional stress-strain relationships, where Lin^+ denotes the space of linear applications between $\mathbb{R}^3$ and $\mathbb{R}^3$ with positive determinant. The elastic potential $\mathcal{W}$ is defined per unit reference volume and represents the path-independent work done by the stresses from initial to current configuration.

A material experiencing finite, i.e. large, strains without perceptible changes in volume is said to be incompressible and is characterized by the constraint $J = 1$. This condition must be introduced into the problem formulation by using the technique of Lagrange multipliers or a penalty method, see, e.g., [MS07]. As commonly done in the treatment of nonlinear elasticity using FEM, we adopt the latter approach, adding a volumetric term to the expression of the strain energy function, that will ultimately result of the form

$$\mathcal{W} = \mathcal{W}(\mathbf{F}) + \mathcal{W}_{vol}(J).$$

In order to penalize large variations of volume, and therefore ensure the incompressibility condition, $\mathcal{W}_{vol}(J)$ is usually chosen to be a convex function in J and such that $J = 1$ is the global minimum. Several expressions for the volumetric function can be found in the literature [DS00], depending on the bulk modulus as penalty term.

1.1.3 Balance of linear momentum

We have now all the ingredients required to formulate the mathematical model to describe the motion of a body in terms of an initial boundary-value problem (IBVP) for the elastodynamics equation, that is, the equation of motion for continuum mechanics.

Let the body $\mathcal{B}$ be a closed system whose mass m is conserved, such that

$$\frac{dm}{dt} = \frac{d}{dt}\int_{\Omega_0} \rho_0(\mathbf{X})dV = \int_{\Omega_0} \dot{\rho}_0(\mathbf{X})dV = 0,$$

where ρ_0 is the (reference) mass density, and define the total linear momentum $\mathbf{L}$ as

$$\mathbf{L}(t) = \int_{\Omega_0} \rho_0(\mathbf{X})\dot{\mathbf{u}}(\mathbf{X}, t)dV,$$

being $\dot{\mathbf{u}}(\mathbf{X}, t) = \partial_t \mathbf{u}(\mathbf{X}, t)$ the material velocity field. The balance of linear momentum states that the change in time of the linear momentum must equal the resultant force acting on the body, i.e.

$$\frac{d}{dt}\int_{\Omega_0} \rho_0(\mathbf{X})\dot{\mathbf{u}}(\mathbf{X}, t)dV = \int_{\partial\Omega_0} \mathbf{T}(\mathbf{X}, t, \mathbf{N})dS + \int_{\Omega_0} \mathbf{b}_0(\mathbf{X}, t)dV,$$

where $\mathbf{b}_0$ is the reference (pseudo[2]) body force, such as gravity loading, and $\mathbf{T}$ is the pseudo traction vector. From the integral form of Cauchy stress theorem (1.1) and employing the divergence theorem, we obtain

$$\int_{\partial\Omega_0} \mathbf{T}(\mathbf{X}, t, \mathbf{N})dS = \int_{\partial\Omega_0} \mathbf{P}(\mathbf{X}, t)\mathbf{N}dS = \int_{\Omega_0} \nabla_0 \cdot \mathbf{P}(\mathbf{X}, t)dV,$$

where $\nabla_0 \cdot (\cdot)$ denotes the material divergence. By replacing this equation in the balance of linear momentum and imposing the conservation of mass, we obtain the Cauchy equation of motion

$$\int_{\Omega_0} \rho_0(\mathbf{X})\ddot{\mathbf{u}}(\mathbf{X}, t)dV = \int_{\Omega_0} \nabla_0 \cdot \mathbf{P}(\mathbf{X}, t)dV + \int_{\Omega_0} \mathbf{b}_0(\mathbf{X}, t)dV$$

for all times t, which can be written in local form as $\rho_0\ddot{\mathbf{u}} = \nabla_0 \cdot \mathbf{P} + \mathbf{b}_0$. This equation is a generalization of Newton second principle of dynamics. The conservation of mass and of linear momentum represent fundamental laws governing the mechanical behavior of a material body.

To conclude, the IBVP for elastodynamics states: given a body force field $\mathbf{b}_0 = \mathbf{b}_0(\mathbf{X}, t)$, a prescribed displacement $\bar{\mathbf{u}} = \bar{\mathbf{u}}(\mathbf{X}, t)$ and surface traction $\bar{\mathbf{T}} = \bar{\mathbf{T}}(\mathbf{X}, t, \mathbf{N})$, find the displacement field $\mathbf{u}\colon \Omega_0 \times (0, T) \to \mathbb{R}^3$ such that

$$\begin{cases} \rho_0\ddot{\mathbf{u}} = \nabla_0 \cdot \mathbf{P} + \mathbf{b}_0 & \text{in } \Omega_0 \times (0, T) \\ \mathbf{u} = \bar{\mathbf{u}} & \text{on } \Gamma_0^D \times (0, T) \\ \mathbf{P}\mathbf{N} = \bar{\mathbf{T}} & \text{on } \Gamma_0^N \times (0, T) \\ \mathbf{P}\mathbf{N} + \alpha\mathbf{u} + \beta\dot{\mathbf{u}} = \mathbf{0} & \text{on } \Gamma_0^R \times (0, T) \\ \mathbf{u} = \mathbf{u}_0, \dot{\mathbf{u}} = \dot{\mathbf{u}}_0 & \text{in } \Omega_0 \times \{0\} \end{cases} \tag{1.2}$$

where $\alpha, \beta \in \mathbb{R}$, $\Gamma_0^D \cup \Gamma_0^N \cup \Gamma_0^R = \partial\Omega_0$ and $\Gamma_0^i \cap \Gamma_0^j = \emptyset$ for $i, j \in \{D, N, R\}$.

In the analysis of continuous media, there exist two main sources of nonlinearity: material nonlinearity, which occurs when the stress-strain relationship is given by nonlinear constitutive equations, and geometric nonlinearity, when the load deformation behavior is highly influenced by changes in geometry, [BW97]. Many problems of engineering and clinical interest, such as the behavior of the cardiac muscle, involve at least one type of nonlinearity, so that we cannot rely on the linear assumption to properly describe the continuum deformation. Numerical analysis, in the form, e.g., of the finite element method (FEM), allows to properly simulate

[2] $\mathbf{b}_0$ acts on the current region Ω_t and is referred to the reference position $\mathbf{X}$, measuring force per unit reference volume.

and investigate the complex nonlinear behavior of solid bodies in a bunch of different scenarios. In the FEM procedure, the continuous body is approximated by a finite number of points, called nodes, and associated regions, called *finite elements* (FE), which form the underlying computational grid. The governing equations of nonlinear continuum mechanics, traditionally recast in a weak integral form, are thus solved by means of suitable numerical schemes onto specific FE points, ultimately yielding to a set of nonlinear algebraic equations, see, e.g., [Cia02; Che05; QV08; EG13]. Further details are given in section 1.3.

1.2 The motivating application: cardiac mechanics

Continuum mechanics represents the fundamental theory for the development of mathematical models characterizing cardiac mechanics and allows for a quantitative analysis of the heart function. In this section we give a brief description of normal cardiac physiology and anatomy, referring, e.g., to [JKT06; Iai09; Kat10; QMV17; TD18] for a full discussion.

1.2.1 Overview of cardiac functions

The human heart is an hollow muscular organ around the size of a closed fist and shaped like an inverted cone, whose main physiological function is to pump blood into the circulatory system. It is separated by the septum into a left and a right part, each made of two chambers, the atrium and the ventricle. Between atria and ventricles and between the left ventricle and the ascending aorta, a set of valves allow the unidirectional blood to flow in the correct direction and prevent backward flows. The heart wall is composed of three tissue layers: the inner *epicardium*, the middle *myocardium* and the outer *epicardium*. The latter is surrounded by the *pericardium*, a fibrous sac that isolates the heart from the closest organs and lubricates its movements, [JKT06].

Due to its crucial function, the left ventricle (LV) represents the most studied among the four cardiac chambers. It is characterized by an ellipsoidal shape, thick muscular myocardium, which can increase its volume by up to 40% during the cardiac cycle [WFC85], and a more irregular interior structure, compared to the right ventricle. The left ventricular myocardium consists of several layers of cardiac muscle fibers, that wrap around the ventricle forming a heterogeneous structure [SJ+69; Gre+81]. This particular fiber architecture of the cardiomyocytes, the excitable cardiac muscle cells, highly influences the mechanical response of the heart, allowing for the sophisticated, yet effective, movements of the LV.

Cardiac muscle fibers contract via excitation-contraction coupling [Ber02], which describes the process of converting an electrical stimulus (action potential) into a mechanical response (muscle contraction). This electrical signal originates at the sinoatrial node, in the upper part of the right atrium, it propagates through the atrial cells to the atrioventricular node, whose purpose is to ensure that the ventricle contraction only starts when the blood has been pumped out of the atria, and finally travels through the Purkinje network, transmitting the cardiac impulse almost simultaneously to the ventricles, from the heart apex to its base. The heart itself is able to produce such electrical impulse, triggered by a transmembrane potential generated by the alternative opening and closing of voltage-gated ion channels (e.g. calcium, sodium and potassium). In particular, calcium ions $[Ca^{2+}]$ induce a complex chain of reactions which generates, at the microscopic level, the active force inside the cardiomyocytes. This results, at

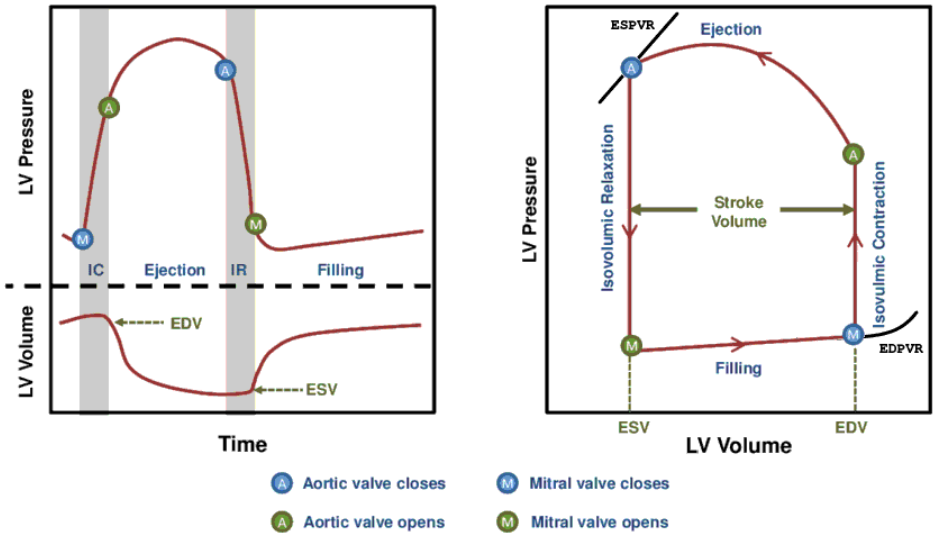


Figure 1.1: LV pressure and volume over time (left) and pressure-volume loop (right), taken from https://www.researchgate.net/figure/The-Pressure-Volume-loop_fig1_281955168

the macroscopic level, in the contraction of the heart chambers, characterized by longitudinal shortening, wall thickening and torsion around the long-axis.

The activity between two consecutive heartbeats is referred to as the cardiac cycle, consisting of alternating contraction, or *systole*, and relaxation, or *diastole*, of the cardiac chambers. In particular, venous blood drains from the systemic circulation into the right atrium, enters through the tricuspid valve in the right ventricle and is pumped into the pulmonary circulation, where it receives oxygen and gives off carbon dioxide. Hence, the oxygenated blood goes back to the left atrium and, through the mitral valve, enters the LV, which finally pumps it out back to the systemic circulation through the aorta, the largest artery of the body.

From a mechanical viewpoint, the cardiac cycle is due to blood movement resulting from pressure differences (or pressure gradient) within the different chambers and blood vessels. The relationship between pressure and volume in the LV over an entire heartbeat represents a very useful way to gain a great insight of cardiac mechanics function [Sag88]. In the so-called pressure-volume loop, see Figure 1.1, the LV cycle is depicted counterclockwise and consists of four phases: diastolic filling, isometric contraction, systolic ejection and isometric relaxation. The end systolic pressure-volume relationship (ESPVR), corresponding to the curve in the upper left part of Figure 1.1, reflects the behavior of the LV during systole due to the active material properties of maximally contracted myocardium. The slope of the ESPVR can be regarded as a useful index of cardiac contractility [Kas+89]: increased contractility will indeed result in a higher slope. Conversely, the curve in the lower right, the end diastolic pressure-volume relationship (EDPVR), is associated with the behavior of the LV during filling and is mainly determined by passive material properties. The EDPVR is nonlinear, as higher and higher pressures are required to continue to fill the ventricle, so that increases in the end diastolic volume leads to increased net amount of blood ejected by the LV, known as stroke volume (SV). The Frank-Starling law expresses the fact that the heart adapts to variations in left ventricular filling, such that, an increase in blood volume, hence in muscle fiber length, determine higher

active tension (total tension minus resting tension), leading to a stronger contraction.

1.2.2 Passive and active mechanics

Cardiac mechanics problems aim at understanding the response of the cardiac muscle under specified loading conditions. However, establishing the stress-strain relationship for biological materials is a complex task. The myocardium is assumed to be hyperelastic, with nonlinear passive behavior and, usually, not isotropic, as experiments have shown higher material stiffness and mechanical response along the cardiac fibers. Many constitutive laws have been derived for both its passive and active description [GM91; GM93; CHM01; HO09; Nor+11; Wan+13], taking into account varying material symmetries. In this Thesis we adopt the law proposed in [GM91] and commonly referred to as the *Guccione* relation, which assumes the material to be transversely isotropic, with its primary material axis oriented along the local fiber direction. The corresponding strain-energy density function is given by

$$\mathcal{W}(\mathbf{F}) = \frac{C}{2}(e^{Q(\mathbf{F})} - 1), \tag{1.3}$$

with the following form for Q to describe three-dimensional transverse isotropy with respect to the fiber coordinate system,

$$Q = b_f E_{ff}^2 + b_s E_{ss}^2 + b_n E_{nn}^2 + b_{fs}(E_{fs}^2 + E_{sf}^2) + b_{fn}(E_{fn}^2 + E_{nf}^2) + b_{sn}(E_{sn}^2 + E_{ns}^2).$$

Here, E_{ij}, $i, j \in \{f, s, n\}$ are the components of the Green-Lagrange strain tensor $\mathbf{E}$, the material constant C scales the stresses and the coefficients b_f, b_s, b_n are related to the material stiffness in the fiber, sheet and cross-fiber directions, respectively.

We point out that, since biological tissues are mostly composed of water, the density ρ_0 is often taken constant in time [Nor+11] so that the conservation of mass implies $J = 1$ and an incompressible formulation should be adopted. However, as often done in cardiac mechanics [Bar+18; DGQ20], incompressibility is weakly imposed, and an isochoric-volumetric decoupling of the strain energy function is employed, yielding

$$\mathcal{W}(\mathbf{F}) = \frac{C}{2}(e^{Q(\mathbf{F})} - 1) + \mathcal{W}_{vol}(J).$$

Another crucial aspect to be taken into account when modeling cardiac material is the inclusion of the active contractile forces in the constitutive equation. Active properties are time-dependent and anisotropic, with more active stress generated along the local muscle fiber direction [LY98]. Active tension can be integrated into the passive stress tensor in different ways, among which the active stress and the active strain represent the most common approaches [AP12]. Here, we adopt the former, and add to the passive first Piola-Kirchhoff stress tensor a time-dependent active tension, which is assumed to act only in the fiber direction

$$\mathbf{P}(\mathbf{F}) = \frac{\partial \mathcal{W}(\mathbf{F})}{\partial \mathbf{F}} + T_a(t)(\mathbf{F}\mathbf{f}_0 \otimes \mathbf{f}_0), \tag{1.4}$$

where $\mathbf{f}_0 \in \mathbb{R}^3$ denotes the reference unit vector in the fiber direction.

1.3 High-fidelity approximation and full order model

We now outline the full order model arising from the high-fidelity approximation of the elastodynamics problem (1.2), which represents the FOM our reduction strategy will be built upon. After the introduction of the weak formulation for the IBVP (1.2), we derive its numerical approximation by relying on the Galerkin-FEM for space discretization and on backward differentiation formula (BDF) schemes for time discretization. This yields a finite set of nonlinear algebraic equations, usually solved by means of the Newton-Raphson iterative technique, also known as Newton method.

1.3.1 Weak formulation

We assume, without loss of generality, homogeneous Dirichlet boundary condition (that is, we set Dirichlet data to $\bar{\mathbf{u}} = \mathbf{0}$) and define the following functional space over $\Omega_0 \subset \mathbb{R}^3$

$$V(\Omega_0) := [H^1_{\Gamma_0^D}(\Omega_0)]^3 = \left\{ \boldsymbol{\eta} \in [H^1(\Omega_0)]^3 \colon \boldsymbol{\eta} = \mathbf{0} \text{ on } \Gamma_0^D \right\}.$$

By multiplying the equation of motion $(1.2)_1$ by a test function $\boldsymbol{\eta} \in V(\Omega_0)$ and integrating over Ω_0, we obtain

$$\begin{aligned} 0 &= \int_{\Omega_0} (\rho_0 \ddot{\mathbf{u}}(t) - \nabla_0 \cdot \mathbf{P}(\mathbf{u}(t)) - \mathbf{b}_0(t)) \cdot \boldsymbol{\eta} d\Omega \\ &= \int_{\Omega_0} \rho_0 \ddot{\mathbf{u}}(t) \cdot \boldsymbol{\eta} d\Omega - \int_{\Omega_0} \nabla_0 \cdot \mathbf{P}(\mathbf{u}(t)) \cdot \boldsymbol{\eta} d\Omega - \int_{\Omega_0} \mathbf{b}_0(t) \cdot \boldsymbol{\eta} d\Omega \\ &= \int_{\Omega_0} \rho_0 \ddot{\mathbf{u}}(t) \cdot \boldsymbol{\eta} d\Omega + \int_{\Omega_0} \mathbf{P}(\mathbf{u}(t)) \colon \nabla \boldsymbol{\eta} d\Omega - \int_{\partial\Omega_0} \mathbf{P}(\mathbf{u}(t)) \cdot \mathbf{N} \cdot \boldsymbol{\eta} d\Gamma - \int_{\Omega_0} \mathbf{b}_0(t) \cdot \boldsymbol{\eta} d\Omega \end{aligned}$$

for $t \in (0, T)$, where we have applied the product rule and the divergence theorem on the second term of the right-hand side. Taking into account the boundary conditions and the fact that $\boldsymbol{\eta} \in V(\Omega_0)$, the weak formulation of (1.2) reads: for $t \in (0, T)$, find the displacement field $\mathbf{u}(t) \in V(\Omega_0)$ such that

$$\begin{aligned} \langle R(\mathbf{u}(t)), \boldsymbol{\eta} \rangle := &\int_{\Omega_0} \rho_0 \ddot{\mathbf{u}}(t) \cdot \boldsymbol{\eta} d\Omega + \int_{\Omega_0} \mathbf{P}(\mathbf{u}(t)) : \nabla \boldsymbol{\eta} d\Omega + \int_{\Gamma_0^R} (\alpha \mathbf{u}(t) + \beta \dot{\mathbf{u}}(t)) \cdot \boldsymbol{\eta} d\Gamma \\ &- \int_{\Gamma_N} \bar{\mathbf{T}}(t, \mathbf{N}) \cdot \boldsymbol{\eta} d\Gamma - \int_{\Omega_0} \mathbf{b}_0(t) \cdot \boldsymbol{\eta} d\Omega = 0 \quad \forall \boldsymbol{\eta} \in V(\Omega_0). \end{aligned} \tag{1.5}$$

For the solution of the nonlinear problem above, a very common strategy is to use the Newton method, which relies on the concept of linearization. In particular, (1.5) is replaced by a sequence of linear problems of the following form: given $\hat{\mathbf{u}} \in V(\Omega_0)$, find $\delta\hat{\mathbf{u}}$ such that

$$\langle J(\hat{\mathbf{u}})[\delta\hat{\mathbf{u}}], \boldsymbol{\eta} \rangle = -\langle R(\hat{\mathbf{u}}), \boldsymbol{\eta} \rangle \qquad \forall \boldsymbol{\eta} \in V(\Omega_0) \tag{1.6}$$

where $J(\hat{\mathbf{u}}) = DR(\hat{\mathbf{u}})$ is the Fréchet differential of R evaluated at $\hat{\mathbf{u}}$. From the linearization of (1.5) around a generic displacement $\hat{\mathbf{u}}$, we found

$$\langle J(\hat{\mathbf{u}})[\delta\hat{\mathbf{u}}], \boldsymbol{\eta} \rangle = \int_{\Omega_0} \rho_0 \ddot{\delta\hat{\mathbf{u}}} \cdot \boldsymbol{\eta} d\Omega + \int_{\Omega_0} \left[\frac{\partial \mathbf{P}(\hat{\mathbf{u}})}{\partial \mathbf{F}} \colon \nabla \delta\hat{\mathbf{u}} \right] : \nabla \boldsymbol{\eta} d\Omega + \int_{\Gamma_0^R} \left(\alpha \delta\hat{\mathbf{u}} + \beta \dot{\delta\hat{\mathbf{u}}} \right) \cdot \boldsymbol{\eta} d\Gamma.$$

1.3.2 Space discretization

We can now introduce the FE discretization starting from the weak formulation (1.5) obtained in previous section. The key idea is to approximate the continuous behavior of $\mathbf{u}(t) \in V(\Omega_0)$ in an element-wise manner on a finite set of elements. With this aim, we introduce a hexahedral mesh $\mathcal{T}_h$ of Ω_0 such that $\bigcup_{\tau\in\mathcal{T}_h} \tau = \Omega_0$, where $h > 0$ denotes the grid size, and define the FE space of dimension $r \geq 1$

$$\mathcal{X}_h^r = \left\{ v \in C^0(\bar{\Omega}_0) \colon v|_\tau \in \mathbb{Q}_r(\tau) \quad \forall \tau \in \mathcal{T}_h \right\},$$

where $\mathbb{Q}_r(\tau)$ is the set of polynomials of degree smaller than or equal to r on each element τ and $\dim(\mathcal{X}_h^r) = N_{h,r}^{dof}$ denotes the total number of degrees of freedom (dofs). The FE space of vector-valued functions is defined as

$$V_h = V(\Omega_0) \cap [\mathcal{X}_h^r]^3 = \operatorname{span}\{\boldsymbol{\varphi}_i\}_{i=1}^{N_h}$$

whose dimension $N_h = 3N_{h,r}^{dof}$ corresponds to the total number of structural dofs, including those associated with the Dirichlet boundary conditions, and $\boldsymbol{\varphi}_i \colon \Omega_0 \to \mathbb{R}^3$, for $i = 1, \dots, N_h$, are the vector-valued basis functions. Given $\mathbf{w}(t) \in V(\Omega_0)$, the corresponding element of V_h can be expressed as a linear combination of the basis functions $\{\boldsymbol{\varphi}_i\}_{i=1}^{N_h}$ as follows

$$\tilde{\mathbf{w}}_h(t) = \sum_{i=1}^{N_h} \mathbf{w}_{h,i}(t)\ \boldsymbol{\varphi}_i,$$

where $\tilde{\mathbf{w}}_h(t) \approx \mathbf{w}(t)$ and $\mathbf{w}_h(t) = [\mathbf{w}_{h,i}(t)]_{i=1}^{N_h} \in \mathbb{R}^{N_h}$ denotes the corresponding vector of coefficients in the expansion with respect to the FE basis, that is, the unknown of our high-fidelity approximation.

We can now introduce the semi-discrete Galerkin-FE approximation of the continuous IBVP, that is the following second-order dynamical system: for each $t \in (0, T)$, find $\mathbf{u}_h(t) \in V_h$ such that

$$\begin{cases} \rho_0 \mathcal{M} \ddot{\mathbf{u}}_h(t) + \mathcal{F}_\beta^{int} \dot{\mathbf{u}}_h(t) + \mathcal{F}_\alpha^{int} \mathbf{u}_h(t) + \mathcal{S}(\mathbf{u}_h(t)) = \mathcal{F}^{ext}(t) \\ \mathbf{u}_h(0) = \mathbf{u}_{h,0}, \quad \dot{\mathbf{u}}_h(0) = \dot{\mathbf{u}}_{h,0} \end{cases}$$

where $\mathbf{u}_{h,0} = [(\mathbf{u}_0, \boldsymbol{\varphi}_i)_{[L^2(\Omega_0)]^3}]_{i=1}^{N_h}$, $\dot{\mathbf{u}}_{h,0} = [(\dot{\mathbf{u}}_0, \boldsymbol{\varphi}_i)_{[L^2(\Omega_0)]^3}]_{i=1}^{N_h}$ and

$$[\mathcal{M}]_{ij} = \int_{\Omega_0} \boldsymbol{\varphi}_j \cdot \boldsymbol{\varphi}_i d\Omega, \qquad [\mathcal{F}_\beta^{int}]_{ij} = \int_{\Gamma_0^R} \beta\ \boldsymbol{\varphi}_j \cdot \boldsymbol{\varphi}_i d\Gamma, \qquad [\mathcal{F}_\alpha^{int}]_{ij} = \int_{\Gamma_0^R} \alpha\ \boldsymbol{\varphi}_j \cdot \boldsymbol{\varphi}_i d\Gamma,$$

$$[\mathcal{S}(\mathbf{u}_h(t))]_i = \int_{\Omega_0} \mathbf{P}(\mathbf{u}_h(t)) \colon \nabla \boldsymbol{\varphi}_i d\Omega, \qquad [\mathcal{F}^{ext}(t)]_i = \int_{\Gamma_0^N} \bar{\mathbf{T}}(t, \mathbf{N}) \cdot \boldsymbol{\varphi}_i d\Gamma + \int_{\Omega_0} \mathbf{b}_0(t) \cdot \boldsymbol{\varphi}_i d\Omega,$$

for all $i, j = 1, \dots, N_h$. The spatially semi-discrete form on the IBVP can thus be equivalently reformulated as: for each $t \in (0, T)$, find $\mathbf{u}_h(t) \in V_h$ such that

$$\widetilde{\mathbf{R}}(\mathbf{u}_h(t), t) := \rho_0 \mathcal{M} \ddot{\mathbf{u}}_h(t) + \mathcal{F}_\beta^{int} \dot{\mathbf{u}}_h(t) + \mathcal{F}_\alpha^{int} \mathbf{u}_h(t) + \mathcal{S}(\mathbf{u}_h(t)) - \mathcal{F}^{ext}(t) = \mathbf{0}. \tag{1.7}$$

We remark that problem (1.7) is a high-dimensional, nonlinear, second-order dynamical system.

For the approximation of cardiac mechanics, quadratic FE ($\mathbb{Q}_2$) are commonly used [Lan+15]. However, due to the huge computational costs entailed, especially when fine meshes are used, in

Figure 1.2: Example of computed tomography scan of a left ventricle (left), the Zygote Solid 3D model [Inc14] (center) and a corresponding mesh with 36704 elements (right). The CT scan has been taken from https://images.app.goo.gl/SGUuFVAQ8RxyDbsw7.

this Thesis we always rely on linear FE, i.e. $r = 1$, which proved to be sufficiently accurate for the purposes at hand and less expensive. We point out that the reduction strategies described in Chapters 2 and 4 are independent of the degree r. To give an idea of the complexity of the problem, in Figure 1.2 we report an example a computational grid $\mathcal{T}_h$ used in this Thesis for the FE discretization, resulting in a total number $N_h \approx 127000$ of dofs when $\mathbb{Q}_1$-FE are employed. The hexahedral mesh has been generated through ad-hoc processing of a CAD geometry (the Zygote Solid 3D heart model [Inc14]) reconstructed from clinical images. We point out that mesh generation is not addressed in this Thesis, and we refer to [FQ21].

1.3.3 Time discretization

For the time discretization of the dynamical system (1.7), we consider a uniform partition $\{t^n = n\Delta t, \quad n = 0, \dots, N_t\}$ of the interval $(0, T)$, where $\Delta t = \frac{T}{N_t}$ is the time step length. To approximate first and second derivatives at time t^n, we employ the following BDF schemes

$$\dot{\mathbf{u}}_h(t^n) \approx \frac{\mathbf{u}_h^n - \mathbf{u}_h^{n-1}}{\Delta t}, \qquad \ddot{\mathbf{u}}_h(t^n) \approx \frac{\mathbf{u}_h^n - 2\mathbf{u}_h^{n-1} + \mathbf{u}_h^{n-2}}{\Delta t^2},$$

respectively, where superscripts n, $n-1$ and $n-2$ denote the solution $\mathbf{u}_h$ computed at time t^n, t^{n-1} and t^{n-2}, e.g. $\mathbf{u}_h(t^n) = \mathbf{u}_h^n$. From now on, we indicate all the quantities computed at time t^n with the superscript n, for $n = 0, \dots, N_t$.

Note that we employ implicit time integration schemes in order to avoid restrictions on the time step due to the highly nonlinear terms of the strain energy density function considered in the numerical examples reported in this Thesis, like the one appearing in (1.3).

Thus, we obtain the following discrete approximation of (1.7): for each $n = 1, \dots, N_h$, find $\mathbf{u}_h^n \in \mathbb{R}^{N_h}$ such that

$$\mathbf{R}(\mathbf{u}_h^n, t^n) = \mathbf{0}, \tag{1.8}$$

where $\mathbf{u}_h^{-1}$ and $\mathbf{u}_h^0$ are given, and

$$\begin{aligned}\mathbf{R}(\mathbf{u}_h^n, t^n) := &\left(\frac{\rho_0}{\Delta t^2}\mathcal{M} + \frac{1}{\Delta t}\mathcal{F}_\beta^{int} + \mathcal{F}_\alpha^{int}\right)\mathbf{u}_h^n + \mathcal{S}(\mathbf{u}_h^n) - \left(\frac{2\rho_0}{\Delta t^2}\mathcal{M} + \frac{1}{\Delta t}\mathcal{F}_\beta^{int}\right)\mathbf{u}_h^{n-1} \\ &+ \frac{\rho_0}{\Delta t^2}\mathcal{M}\mathbf{u}_h^{n-2} - \mathcal{F}^{ext,n}.\end{aligned} \tag{1.9}$$

1.3.4 Fully discretized Newton system

In this section we first recall the Newton method for the solution of a nonlinear algebraic system of the form: find $\mathbf{w} \in \mathbb{R}^{N_h}$ such that

$$\mathbf{R}(\mathbf{w}) = \mathbf{0},$$

and then apply it to system (1.8).

Starting from an initial guess $\mathbf{w}^{(0)}$, at each iteration $k \geq 0$, given $\mathbf{w}^{(k)}$, we obtain the value $\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} + \delta\mathbf{w}^{(k)}$ from the solution of the linearized system

$$\mathbf{R}(\mathbf{w}^{(k+1)}) \approx \mathbf{R}(\mathbf{w}^{(k)}) + D\mathbf{R}(\mathbf{w}^{(k)})[\delta\mathbf{w}^{(k)}] = \mathbf{0}, \tag{1.10}$$

until a suitable stopping criterion is fulfilled. Here, $D\mathbf{R}(\mathbf{w}^{(k)})[\delta\mathbf{w}^{(k)}]$ denotes the directional (or Gâteaux) derivative of $\mathbf{R}$ at $\mathbf{w}^{(k)}$ in the direction of an increment $\delta\mathbf{w}^{(k)}$, which can be computed as

$$D\mathbf{R}(\mathbf{w}^{(k)})[\delta\mathbf{w}^{(k)}] = \frac{d}{d\boldsymbol{\varepsilon}}\Big|_{\boldsymbol{\varepsilon}=0}\mathbf{R}(\mathbf{w}^{(k)} + \boldsymbol{\varepsilon}\delta\mathbf{w}^{(k)}) = \mathbf{J}(\mathbf{w}^{(k)})\delta\mathbf{w}^{(k)},$$

so that $\mathbf{J}_{ij}(\mathbf{w}^{(k)}) = \frac{\partial \mathbf{R}_i}{\partial \mathbf{w}_j}(\mathbf{w}^{(k)})$. It is important to note that, although $\mathbf{R}$ is nonlinear in $\mathbf{w}$, $D\mathbf{R}(\mathbf{w}^{(k)})[\delta\mathbf{w}^{(k)}]$ is always linear in $\delta\mathbf{w}^{(k)}$. Thus, we can rewrite equation (1.10) as

$$\mathbf{J}(\mathbf{w}^{(k)})\delta\mathbf{w}^{(k)} = -\mathbf{R}(\mathbf{w}^{(k)}).$$

The computation of the Jacobian (tangent stiffness matrix) $\mathbf{J}$ arising from the space-time discretization of (1.6) is a complex task, for which in this Thesis we rely on the useful tool of automatic differentiation (AD) provided in the `deal.II` (https://www.dealii.org) Finite Element library [Arn+20].

To conclude, when applied to the solution of (1.9), the Newton method reads as follows: for each $n = 1, \ldots, N_t$, given $\mathbf{u}_h^{n,(0)} = \mathbf{u}_h^{n-1}$, for $k \geq 0$, find $\delta\mathbf{u}_h^{(k)} \in \mathbb{R}^{N_h}$ such that

$$\begin{cases} \mathbf{J}(\mathbf{u}_h^{n,(k)}, t^n)\delta\mathbf{u}^{(k)} = -\mathbf{R}(\mathbf{u}^{n,(k)}, t^n) \\ \mathbf{u}^{n,(k+1)} = \mathbf{u}^{n,(k)} + \delta\mathbf{u}^{(k)} \end{cases}$$

until $\|\mathbf{R}(\mathbf{u}^{n,(k+1)}, t^n)\|_2 / \|\mathbf{R}(\mathbf{u}^{n,(0)}, t^n)\|_2 < \varepsilon_{Nwt}$, where $\varepsilon_{Nwt} > 0$ is a prescribed tolerance. Note that, for $n = 1$, the initial guess is given by the initial condition, i.e. $\mathbf{u}_h^{1,(0)} = \mathbf{u}_{h,0}$.

Broyden method

As an alternative to Newton iterative scheme, we can rely on Broyden's method [Bro65], which belongs to the class of quasi-Newton methods, as done, e.g, in [MBQ18; RR16] for model order reduction of nonlinear elasticity problems.

The key idea is to avoid the computation of the Jacobian matrix $\mathbf{J}(\mathbf{w}^{(k)})$ at each iteration $k \geq 0$ by replacing it with a suitable approximation $\widetilde{\mathbf{J}}^{(k)}$. Therefore, given $\widetilde{\mathbf{J}}^{(0)}$, for $k \geq 0$ until convergence, one must solve the system

$$\begin{cases} \widetilde{\mathbf{J}}^{(k)}\delta\mathbf{w}^{(k)} = -\mathbf{R}(\mathbf{w}^{(k)}) \\ \mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} + \delta\mathbf{w}^{(k)} \end{cases} \tag{1.11}$$

and set

$$\widetilde{\mathbf{J}}^{(k+1)} = \widetilde{\mathbf{J}}^{(k)} + \frac{(\mathbf{R}(\mathbf{w}^{(k+1)}) - \mathbf{R}(\mathbf{w}^{(k)})) - \widetilde{\mathbf{J}}^{(k)}\delta\mathbf{w}^{(k)}}{(\delta\mathbf{w}^{(k)})^T(\delta\mathbf{w}^{(k)})}(\delta\mathbf{w}^{(k)})^T,$$

until $\|\mathbf{R}(\mathbf{w}^{(k+1)})\|_2/\|\mathbf{R}(\mathbf{w}^{(0)})\|_2 < \varepsilon_{Brdn}$. For the initialization of (1.11), a common choice for $\widetilde{\mathbf{J}}^{(0)}$ is the Jacobian matrix $\mathbf{J}(\mathbf{w}^{(0)})$, however, other approximations are also possible.

When applied to (1.9), the Broyden method can be written as follows: for each $n = 1, \dots, N_t$, given $\mathbf{u}_h^{n,(0)} = \mathbf{u}_h^{n-1}$ and an initial guess $\widetilde{\mathbf{J}}(\mathbf{u}_h^{n,(0)}, t^n)$ for the Jacobian, for $k \geq 0$, find $\delta\mathbf{u}_h^{(k)} \in \mathbb{R}^{N_h}$ such that

$$\begin{cases} \widetilde{\mathbf{J}}(\mathbf{u}_h^{n,(k)}, t^n)\delta\mathbf{u}^{(k)} = -\mathbf{R}(\mathbf{u}^{n,(k)}, t^n) \\ \mathbf{u}^{n,(k+1)} = \mathbf{u}^{n,(k)} + \delta\mathbf{u}^{(k)} \\ \widetilde{\mathbf{J}}(\mathbf{u}_h^{n,(k+1)}, t^n) = \widetilde{\mathbf{J}}(\mathbf{u}_h^{n,(k)}, t^n) + \dfrac{(\mathbf{R}(\mathbf{u}^{n,(k+1)}, t^n) - \mathbf{R}(\mathbf{u}^{n,(k)}, t^n)) - \widetilde{\mathbf{J}}(\mathbf{u}_h^{n,(k)}, t^n)\delta\mathbf{u}^{(k)}}{(\delta\mathbf{u}^{(k)})^T(\delta\mathbf{u}^{(k)})}(\delta\mathbf{u}^{(k)})^T \end{cases}$$

until $\|\mathbf{R}(\mathbf{u}^{n,(k+1)}, t^n)\|_2/\|\mathbf{R}(\mathbf{u}^{n,(0)}, t^n)\|_2 < \varepsilon_{Brdn}$, where $\varepsilon_{Brdn} > 0$ is a prescribed tolerance.

We point out that, when using AD, as done in our HPC library `lifex`, the computational burden is represented by the assembling of residual vectors, rather than the calculation of the Jacobian matrix. Moreover, being a quasi-Newton method, the Broyden method presents a lower convergence rate than the exact Newton method. For these reasons, in the numerical examples considered in this Thesis, we rely exclusively on the Newton method.

1.3.5 Parameterized full order model

The main purpose of this Thesis is the construction of efficient, yet accurate, reduced order models for the solution of nonlinear elastodynamics problems in different scenarios, with the particular application to cardiac mechanics. The underlying partial differential equation, as well as the boundary and initial conditions or the geometrical domain, might depend on a set of input parameters that may have a great impact on the problem under investigation. We denote the input vector of varying parameters by $\boldsymbol{\mu} \in \mathcal{P} \subset \mathbb{R}^P$, where $\mathcal{P}$ is a suitable closed subset. Thus, we write the parameterized algebraic system (1.8) as

$$\mathbf{R}(\mathbf{u}_h^n(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}) = \mathbf{0}, \quad n = 1, \dots, N_t. \tag{1.12}$$

The Newton method applied to the parameterized nonlinear system (1.12) reads: given an initial guess $\mathbf{u}_h^{n,(0)}(\boldsymbol{\mu})$, for $k \geq 0$, find $\delta\mathbf{u}_h^{(k)}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ such that

$$\begin{cases} \mathbf{J}(\mathbf{u}_h^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu})\delta\mathbf{u}_h^{(k)}(\boldsymbol{\mu}) = -\mathbf{R}(\mathbf{u}_h^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}) \\ \mathbf{u}_h^{n,(k+1)}(\boldsymbol{\mu}) = \mathbf{u}_h^{n,(k)}(\boldsymbol{\mu}) + \delta\mathbf{u}_h^{(k)}(\boldsymbol{\mu}) \end{cases} \tag{1.13}$$

until $\|\mathbf{R}(\mathbf{u}_h^{n,(k+1)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu})\|_2 / \|\mathbf{R}(\mathbf{u}_h^{n,(0)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu})\|_2 < \varepsilon_{Nwt}$, where $\varepsilon_{Nwt} > 0$ is a prescribed tolerance. Here, $\mathbf{u}_h^{n,(0)}(\boldsymbol{\mu})$ is set equal to the initial guess $\mathbf{u}_{h,0}(\boldsymbol{\mu})$, when $n = 1$, and is equal to the solution at previous time iteration $\mathbf{u}_h^{n-1}(\boldsymbol{\mu})$, for $n = 2, \ldots, N_t$. From now on, we will refer to (1.13) as to the FOM for our problem, and denote by N_h the high-fidelity dimension.

In the context of cardiac mechanics, the FOM dimension N_h may become extremely large (of the order of hundreds of thousands) in order to correctly capture the deformation of the tissue, especially when considering detailed geometries or pathological scenarios. Furthermore, if one is interested in the repeated solution to the forward problem for different values of the parameter, the computational cost becomes prohibitive. The development of accurate and efficient reduced order models is thus of paramount importance, both to address the solution to multi-query problems or to allow the use of numerical simulations in clinical practice, and is the subject of Chapters 2 and 4.

Chapter 2

Reduced order modeling for nonlinear parameterized elastodynamics problems

Relying on full order models for the solution of elastodynamics problems, as those introduced in Chapter 1, results in high computational demands. To address this issue, we focus on projection-based model order reduction for nonlinear, time-dependent parameterized PDEs. After a brief introduction of the Galerkin-reduced basis method, we describe the discrete empirical interpolation method used for efficiently handle the nonlinear terms. The accuracy and speed-up of the reduction strategy are assessed in several numerical experiments, while drawbacks and future perspectives are reported at the end of the chapter.

2.1 Solution-space reduction: the reduced basis method

Before dealing with model order reduction of nonlinear, time-dependent parameterized problems, we introduce the reduced basis (RB) method in the case of stationary PDEs characterized by an affine parametric dependence. In this way, we provide all the necessary tools to understand the reduction procedure, before restoring to the more complex case at hand. With this aim, let V_h be a N_h-dimensional space and define

$$u_h \colon \mathcal{P} \to V_h$$

the (discrete) solution operator of a parameterized, stationary PDE. The goal of the RB method for parameterized PDEs is to approximate the N_h-dimensional solution manifold $\mathcal{M}_h = \mathrm{Im}(u_h)$ with a low number N (less than a few dozens, or hundreds at most) of basis functions, forming the so-called *reduced basis*. This is usually done by performing a Galerkin (or, more generally, a Petrov-Galerkin) projection of the high-fidelity problem onto the N-dimensional subspace spanned by the reduced basis, obtaining a reduced problem facing a much lower computational complexity, still respecting the structure of the underlying PDE and retaining the essential features of the parameter-to-solution map.

The efficiency of the RB method relies, mainly, on two assumptions. First, that the solution manifold $\mathcal{M}_h$ is of low-dimension; second, that the reduction procedure can be split into *offline* and *online* stages, where the latter is completely independent of the high-fidelity dimension [Far+20]. The first hypothesis concerns the approximability of the solution set and is associated with the concept of Kolmogorov N-width [Pin85]: if the N-width decays rapidly, then the solution manifold is likely to be well approximated by a small reduced basis. The second assumption is verified in the case of problems featuring an affine parametric dependence [QMN16]. However, a full offline-online decoupling is often not possible in the case of nonlinear problems, so that a further level of approximation must be introduced (see Section 2.2).

2.1.1 Introduction to the RB method

For the sake of simplicity, in this section we introduce the RB method for a coercive and affinely decomposed problem, and later extend the methodology to the class of problems we are interested in.

Let us consider the variational formulation of a parameterized, elliptic, scalar PDE: given the parameter vector $\boldsymbol{\mu} \in \mathcal{P}$, find the solution $u(\boldsymbol{\mu}) \in V$ such that

$$a(u, v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}) \quad \forall v \in V,$$

where $a(\cdot,\ \cdot; \boldsymbol{\mu})\colon V \times V \to \mathbb{R}$ is a symmetric, continuous, coercive bilinear form encoding the differential operator and $f(\cdot; \boldsymbol{\mu})\colon V \to \mathbb{R}$ is a linear continuous functional. Here, $V = V(\Omega_0)$ is a suitable Hilbert space on the bounded domain $\Omega_0 \subset \mathbb{R}$, usually $H_0^1(\Omega_0) \subseteq V \subseteq H^1(\Omega_0)$. Let $V_h \subset V$ be a N_h-dimensional subset of V, endowed with the norm $\|\cdot\|_V$, such that the Galerkin high-fidelity approximation reads: find the solution $u_h(\boldsymbol{\mu}) \in V_h$ of

$$a(u_h, v_h; \boldsymbol{\mu}) = f(v_h; \boldsymbol{\mu}) \quad \forall v_h \in V_h. \tag{2.1}$$

We point out that we are adopting a Galerkin approach, where the trial space and the test space coincide. However, a Petrov-Galerkin approach, where $v_h \in W_h$ with $W_h \neq V_h$, is also possible. By denoting with $\{\phi_j\}_{j=1}^{N_h}$ a basis for V_h, we can write problem (2.1) in the equivalent algebraic form as: find $\mathbf{u}_h(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ such that

$$\mathbb{A}_h(\boldsymbol{\mu})\mathbf{u}_h(\boldsymbol{\mu}) = \mathbf{f}_h(\boldsymbol{\mu}),$$

where $\mathbb{A}_h(\boldsymbol{\mu}) \in \mathbb{R}^{N_h \times N_h}$ is the $\boldsymbol{\mu}$-dependent stiffness matrix whose components are given by $(\mathbb{A}_h(\boldsymbol{\mu}))_{i,j} = a(\phi_j, \phi_i; \boldsymbol{\mu})$, for $i, j = 1, \dots, N_h$, the right-hand side is $\mathbf{f}_h(\boldsymbol{\mu}) = [f(\boldsymbol{\phi}_j; \boldsymbol{\mu})]_{j=1}^{N_h} \in \mathbb{R}^{N_h}$ and $\mathbf{u}_h(\boldsymbol{\mu})$ is the vector of unknown nodal values of $u_h(\boldsymbol{\mu}) = \sum_{j=1}^{N_h} u_h^{(j)}(\boldsymbol{\mu})\phi_j$, that is,

$$\mathbf{u}_h(\boldsymbol{\mu}) = \left[u_h^{(1)}(\boldsymbol{\mu}), \dots, u_h^{(N_h)}(\boldsymbol{\mu})\right]^T.$$

To mitigate the computational costs associated with the solution to the high-fidelity problem, we introduce the following reduction strategy. Consider the solution manifold

$$\mathcal{M}_h = \{u_h(\boldsymbol{\mu}) \in V_h \mid \boldsymbol{\mu} \in \mathcal{P}\} \subset V_h$$

made by all FOM solutions under variation of the parameter vector $\boldsymbol{\mu}$ over the parameter domain $\mathcal{P}$, and assume that it has low-dimension, so that is possible to approximate the high-fidelity solutions as a linear combination of few reduced modes with a small error. We aim at generating an N-dimensional subspace $V_N \subset V_h$, with $N \ll N_h$, in order to replace (2.1) with the following reduced problem: find $u_N(\boldsymbol{\mu}) \in V_N$ such that

$$a(u_N, v_N; \boldsymbol{\mu}) = f(v_N; \boldsymbol{\mu}) \quad \forall v_N \in V_N, \tag{2.2}$$

or, equivalently: find $\mathbf{u}_N(\boldsymbol{\mu}) \in \mathbb{R}^N$ solution to

$$\mathbb{A}_N(\boldsymbol{\mu})\mathbf{u}_N(\boldsymbol{\mu}) = \mathbf{f}_N(\boldsymbol{\mu}).$$

More precisely, we need to:

1. construct a basis $\{\xi_j\}_{j=1}^N$ for V_N, starting from a set of FOM solution *snapshots*

$$S_h = \{u_h(\boldsymbol{\mu}_1), \dots, u_h(\boldsymbol{\mu}_{n_s})\}$$

   obtained for different parameter values $\boldsymbol{\mu}_\ell \in \mathcal{P}$, $\ell = 1, \dots, n_s$, suitably sampled over the parameter domain $\mathcal{P}$, and define the corresponding basis matrix $\mathbf{V} \in \mathbb{R}^{N_h \times N}$;

2. given $\boldsymbol{\mu} \in \mathcal{P}$, set the reduced problem (2.2) via a projection approach;

3. compute the unknown coefficients of $u_N(\boldsymbol{\mu}) = \sum_{j=1}^N u_N^{(j)}(\boldsymbol{\mu})\xi_j \in V_N$ in order to find a reduced approximation to $u_h(\boldsymbol{\mu}) \in V_h$, such that

$$\mathbf{u}_h(\boldsymbol{\mu}) \approx \mathbf{V}\mathbf{u}_N(\boldsymbol{\mu}).$$

Step 1 is done during the costly offline stage of the reduction procedure, which is however performed once and for all, while steps 2 and 3 are performed inexpensively online for every new instance of the parameter vector. Regarding step 2, a classical procedure to obtain the RB solution consists in requiring the fulfillment of a suitable geometric orthogonality criterion, yielding to the Galerkin-RB method. By forcing the high-fidelity residual $\mathbf{r}_h(\mathbf{v}_h; \boldsymbol{\mu}) := \mathbf{f}_h(\boldsymbol{\mu}) - \mathbb{A}_h(\boldsymbol{\mu})\mathbf{v}_h(\boldsymbol{\mu})$ computed over the RB solution to be orthogonal to the subspace V_N, we obtain

$$\begin{aligned}\mathbf{0} &= \mathbf{V}^T \mathbf{r}_h(\mathbf{V}\mathbf{u}_N; \boldsymbol{\mu}) \\ &= \mathbf{V}^T \mathbf{f}_h(\boldsymbol{\mu}) - \mathbf{V}^T \mathbb{A}_h(\boldsymbol{\mu})\mathbf{V}\mathbf{u}_N(\boldsymbol{\mu}) =: \mathbf{f}_N(\boldsymbol{\mu}) - \mathbb{A}_N(\boldsymbol{\mu})\mathbf{u}_N(\boldsymbol{\mu}),\end{aligned}$$

that is, we set

$$\begin{aligned}\mathbb{A}_N(\boldsymbol{\mu}) &= \mathbf{V}^T \mathbb{A}_h(\boldsymbol{\mu})\mathbf{V} \\ \mathbf{f}_N(\boldsymbol{\mu}) &= \mathbf{V}^T \mathbf{f}_h(\boldsymbol{\mu}).\end{aligned} \tag{2.3}$$

Two popular methods for the construction of the RB basis are the greedy algorithm [Edm71], based on an a posteriori error estimator, and the proper orthogonal decomposition (POD) [Lum67], based on a singular value decomposition (SVD) of the snapshots matrix. Since for time-dependent, nonlinear problems error bounds are usually extremely difficult to obtain, in this thesis we rely on the POD method, which Section 2.1.2 is devoted to.

As already mentioned, a convenient situation in view of a computationally efficient offline-online decoupling arises when the N_h-dimensional arrays can be written as linear combinations of (possibly few) $\boldsymbol{\mu}$-independent terms; in the linear steady case, this means that $\mathbb{A}_h(\boldsymbol{\mu})$ and $\mathbf{f}_h(\boldsymbol{\mu})$ can be expressed as

$$\mathbb{A}_h(\boldsymbol{\mu}) = \sum_{q=1}^{Q_a} \theta_q^a(\boldsymbol{\mu})\mathbb{A}_{h,q} \quad \text{and} \quad \mathbf{f}_h(\boldsymbol{\mu}) = \sum_{q'=1}^{Q_f} \theta_{q'}^f(\boldsymbol{\mu})\mathbf{f}_{h,q'}, \tag{2.4}$$

where $\mathbb{A}_{h,q} \in \mathbb{R}^{N_h \times N_h}$ and $\mathbf{f}_{h,q'} \in \mathbb{R}^{N_h}$ are $\boldsymbol{\mu}$-independent arrays, while $\theta_q^a, \theta_{q'}^f : \mathcal{P} \to \mathbb{R}$ are $\boldsymbol{\mu}$-dependent coefficients to be computed online. This affine parametric dependence allows to assemble online the reduced system in a way that is independent of N_h. Indeed, by replacing

(2.4) in (2.3), we obtain

$$\mathbb{A}_N(\boldsymbol{\mu}) = \mathbf{V}^T\mathbb{A}_h(\boldsymbol{\mu})\mathbf{V} = \sum_{q=1}^{Q_a}\theta_q^a(\boldsymbol{\mu})\mathbf{V}^T\mathbb{A}_{h,q}\mathbf{V} = \sum_{q=1}^{Q_a}\theta_q^a(\boldsymbol{\mu})\mathbb{A}_{N,q},$$
$$\mathbf{f}_N(\boldsymbol{\mu}) = \mathbf{V}^T\mathbf{f}_h(\boldsymbol{\mu}) = \sum_{q'=1}^{Q_f}\theta_{q'}^f(\boldsymbol{\mu})\mathbf{V}^T\mathbf{f}_{h,q'} = \sum_{q'=1}^{Q_f}\theta_{q'}^f(\boldsymbol{\mu})\mathbf{f}_{N,q'},$$

so that the reduced arrays $\mathbb{A}_{N,q} = \mathbf{V}^T\mathbb{A}_{h,q}\mathbf{V} \in \mathbb{R}^{N\times N}$ and $\mathbf{f}_{N,q'} = \mathbf{V}^T\mathbf{f}_{h,q'} \in \mathbb{R}^N$ can be computed and stored during the offline stage. Thus, for every new instance of the parameter $\boldsymbol{\mu} \in \mathcal{P}$, we only need to deal with (few) low-dimensional arrays, enabling a vary rapid assembling of the reduced system. The reduction procedure for affinely parameterized PDEs presented so far is summarized in Algorithm 1.

Algorithm 1 Galerkin-RB method for affine problems

Offline stage

INPUT: $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_{n_s} \in \mathcal{P}$

OUTPUT: $\mathbb{A}_{N,q} \in \mathbb{R}^{N\times N}$, for $q = 1, \dots, Q_a$, and $\mathbf{f}_{N,q'} \in \mathbb{R}^N$, for $q' = 1, \dots, Q_f$

1: Generate FOM solution snapshots $\mathbf{u}_h(\boldsymbol{\mu}_\ell)$, for $j = 1, \dots, n_s$
2: Construct RB basis $\mathbf{V} \in \mathbb{R}^{N_h\times N}$
3: Project FOM over RB subspace: $\mathbb{A}_{N,q} = \mathbf{V}^T\mathbb{A}_{h,q}\mathbf{V}$ and $\mathbf{f}_{N,q'} = \mathbf{V}^T\mathbf{f}_{h,q'}$

Online stage

INPUT: $\boldsymbol{\mu} \in \mathcal{P}$

OUTPUT: $\mathbf{V}\mathbf{u}_N(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$

1: Assemble RB system: $\mathbb{A}_N(\boldsymbol{\mu}) = \sum_{q=1}^{Q_a}\theta_q^a(\boldsymbol{\mu})\mathbb{A}_{N,q}$, $\mathbf{f}_N(\boldsymbol{\mu}) = \sum_{q'=1}^{Q_f}\theta_{q'}^f(\boldsymbol{\mu})\mathbf{f}_{N,q'}$
2: Solve $\mathbb{A}_N(\boldsymbol{\mu})\mathbf{u}_N(\boldsymbol{\mu}) = \mathbf{f}_N(\boldsymbol{\mu})$
3: Recover $\mathbf{V}\mathbf{u}_N(\boldsymbol{\mu})$

2.1.2 Reduced basis construction

Due to its ease of implementation and its deep mathematical root (related to the analysis of compact operators, to matrix singular value decomposition and to dimensionality reduction in data analysis, just to mention a few links), POD has gained remarkable success in the scientific computing community and has been applied in a broad range of engineering fields [Aub91; Lia+02; QMN16] to reduce, in an optimal sense, the dimension of a given data set. In this Thesis POD is used to build the RB basis $\mathbf{V}$, as well as for the construction of the DEIM basis $\boldsymbol{\Phi}_{\mathcal{R}}$ for the nonlinear terms (see Section 2.2).

Let $s_h : \mathcal{P} \to V_h$ be a parameter-dependent function, e.g., the solution operator u_h, and $\mathbf{s}_h(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ be the corresponding vector of dofs, where $\dim(V_h) = N_h$. For a given set of training parameters $\mathcal{S} = \{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_{n_s}\} \subset \mathcal{P}$, with $n_s < N_h$, we define the snapshots matrix

$$\mathbf{S} = [\mathbf{s}_1 \mid \dots \mid \mathbf{s}_{n_s}] \in \mathbb{R}^{N_h\times n_s},$$

where $\mathbf{s}_i = \mathbf{s}(\boldsymbol{\mu}_i)$, $i = 1, \dots, n_s$. Note that the training parameters can be randomly selected or sampled using more ad-hoc strategies, such as the *latin hypercube sampling*, especially when

the dimension of the parameter space gets large. POD aims at approximating the manifold identified by the image of s_h, i.e.

$$\mathcal{M}_s = \{s_h(\boldsymbol{\mu}) \in V_h \mid \boldsymbol{\mu} \in \mathcal{P}\},$$

with a low-dimensional linear subspace, retaining as much as possible of the information gathered in the snapshots. To achieve this goal, the singular value decomposition (SVD) of $\mathbf{S}$,

$$\mathbf{S} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{Z}^T,$$

is computed, where $\mathbf{U} = [\boldsymbol{\xi}_1|\dots|\boldsymbol{\xi}_{N_h}] \in \mathbb{R}^{N_h \times N_h}$ and $\mathbf{Z} = [\boldsymbol{\zeta}_1|\dots|\boldsymbol{\zeta}_{n_s}] \in \mathbb{R}^{n_s \times n_s}$ are orthogonal matrices collecting column-wise the left and the right singular vectors, respectively. The diagonal matrix

$$\boldsymbol{\Sigma} = diag(\sigma_1, \dots, \sigma_r) \in \mathbb{R}^{N_h \times n_s}$$

contains all singular values of $\mathbf{S}$, sorted in descending order $\sigma_1 \geq \dots \geq \sigma_r > 0$, where $r \leq \min(N_h, n_s)$ is the rank of $\mathbf{S}$, so that we can write

$$\mathbf{S} = \sum_{j=1}^{r} \sigma_j \boldsymbol{\xi}_j \boldsymbol{\zeta}_j^T.$$

The N-dimensional POD basis is then obtained by collecting the first N columns of $\mathbf{U}$

$$\mathbf{V} = [\boldsymbol{\xi}_1 \mid \dots \mid \boldsymbol{\xi}_N],$$

corresponding to the N largest singular values (see Algorithm 2). At the basis of the POD algorithm stands the Schmidt-Eckart-Young theorem [Sch89; EY36], originally introduced for integral operators, which provides a criterion for the best low-rank approximation of a given matrix of rank r. In fact, the basis $\mathbf{V} \in \mathbf{R}^{N_h \times N}$ is such that

$$\mathbf{V}\mathbf{V}^T\mathbf{S} = \sum_{j=1}^{N} \sigma_j \boldsymbol{\xi}_j \boldsymbol{\zeta}_j^T$$

has rank $N \leq r$ and satisfies the following optimality property:

$$\|\mathbf{S} - \mathbf{V}\mathbf{V}^T\mathbf{S}\|_F^2 = \min_{\substack{\mathbf{A} \in \mathbb{R}^{N_h \times N_h} \\ \text{rank}(\mathbf{A}) \leq N}} \|\mathbf{S} - \mathbf{A}\|_F^2 = \sum_{j=N+1}^{r} \sigma_j^2, \tag{2.5}$$

where $\|\cdot\|_F$ is the Frobenius norm. Thus, span($\mathbf{V}$) is optimal in the sense that, among all possible N-dimensional subspaces, minimizes the least square error of snapshot reconstruction. Furthermore, thanks to (2.5), the singular values of the snapshots matrix provide an heuristic criteria for choosing the basis dimension N, which can be computed as the minimum integer satisfying the condition

$$RIC(N) = \frac{\sum_{i=1}^{N} \sigma_i^2}{\sum_{i=1}^{r} \sigma_i^2} \geq 1 - \varepsilon_{POD}^2, \tag{2.6}$$

where $\varepsilon_{POD} > 0$ is a given tolerance. The left-hand side of (2.6) is the relative information content (RIC) and represents the ratio between the modeled and the total snapshots information content.

Algorithm 2 Proper orthogonal decomposition (POD)

INPUT: $\mathbf{S} \in \mathbb{R}^{N_h \times n_s}$
OUTPUT: $\mathbf{V} \in \mathbb{R}^{N_h \times N}$

1: Perform SVD of $\mathbf{S}$, i.e., $\mathbf{S} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{Z}^T$
2: Select basis dimension N as the minimum integer fulfilling condition (2.6)
3: Construct $\mathbf{V}$ collecting the first N columns of $\mathbf{U}$

The POD technique is efficient, provided that a sufficiently large number of snapshots has been chosen in order to cover the manifold $\mathcal{M}_s$. Furthermore, it can be used as a tool to assess the reducibility of the underlying problem, that is, if the manifold $\mathcal{M}_s$ is likely to be well approximated by a small reduced basis. A rapid decay of the singular values means that a limited number of POD modes is potentially sufficient to represent the entire manifold, hence, the problem is reducible.

Remark 2.1
An efficient, non-deterministic, version of POD can be obtained by relying on the so-called randomized-SVD, see Algorithm 3. Randomization offers, in fact, a powerful tool for performing low-rank matrix approximation, especially when dealing with massive data sets. The randomized approach usually beats its classical competitors in terms of computational speed-up, accuracy and robustness [HMT11]. The key idea of randomized-SVD is to split the task of computing an approximated singular value decomposition of a given matrix into a first random stage and a second deterministic one. The former exploits random sampling to construct a low-dimensional subspace that captures most of the action of the input matrix; the latter is meant to restrict the given matrix to this subspace and then manipulate the associated reduced matrix with classical deterministic algorithms, to obtain the desired low-rank approximations. This randomized approach is particularly convenient when the snapshots matrix is high-dimensional, i.e. when N_h and n_s are large. In fact, finding the first k dominant singular-values for a dense input matrix of dimension $N_h \times n_s$, requires $O(N_h n_s \log(k))$ floating-point operations for a randomized algorithm, in contrast with $O(N_h n_s k)$ flops for a classical one.

Algorithm 3 Randomized-SVD

INPUT: $\mathbf{S} \in \mathbb{R}^{m \times n}$, target rank $k \in \mathbb{N}$
OUTPUT: $\mathbf{U}\boldsymbol{\Sigma}\mathbf{Z}^T \approx \mathbf{S}$

stage 1

1: Generate a Gaussian matrix $\boldsymbol{\Theta} \in \mathbb{R}^{N_h \times k}$
2: Compute $\mathbf{Q} \in \mathbb{R}^{N_h \times k}$ whose columns form an orthonormal basis for the range of $\mathbf{S}\boldsymbol{\Theta}$ and such that

$$\|\mathbf{S} - \mathbf{Q}\mathbf{Q}^T\mathbf{S}\|_2 \leq \min_{rank(\mathbf{X}) \leq k} \|\mathbf{S} - \mathbf{X}\|_2,$$

e.g., using the QR factorization.

stage 2

1: Form $\widetilde{\mathbf{S}} = \mathbf{Q}^T\mathbf{S} \in \mathbb{R}^{k \times n}$
2: Compute SVD of $\widetilde{\mathbf{S}} = \widetilde{\mathbf{U}}\boldsymbol{\Sigma}\mathbf{Z}^T$
3: Set $\mathbf{U} = \mathbf{Q}\widetilde{\mathbf{U}}$

Finally, Algorithm 3 can be adapted to solve the following problem: given a target error tolerance $\varepsilon_{rSVD} > 0$, find $k = k(\varepsilon_{rSVD})$ and $\mathbf{Q} \in \mathbb{R}^{N_h \times k}$ satisfying

$$\|\mathbf{S} - \mathbf{Q}\mathbf{Q}^T\mathbf{S}\|_2 \leq \varepsilon_{rSVD}.$$

2.1.3 RB method for time-dependent nonlinear parameterized PDEs

Based on the idea of the previous section, we extend the RB method to the class of time-dependent, nonlinear problems, whose numerical approximation can be written as follows: given $\boldsymbol{\mu} \in \mathcal{P}$ and $\mathbf{u}_h^0(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$, for each $n = 1, \dots, N_t$, find $\mathbf{u}_h^n(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ such that

$$\mathbf{R}(\mathbf{u}_h^n(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}) = \mathbf{0}.$$

Relying on implicit time schemes requires the solution to as many linear systems as the number of Newton iterations per each time step.

Let us introduce the sample set $\mathcal{S} = \{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_{n_s}\}$, for some $\boldsymbol{\mu}_\ell \in \mathcal{P}$, and define the corresponding snapshots matrix

$$\mathbf{S}_u = \left[\mathbf{u}_h^1(\boldsymbol{\mu}_1) \mid \dots \mid \mathbf{u}_h^{N_t}(\boldsymbol{\mu}_1) \mid \dots \mid \mathbf{u}_h^1(\boldsymbol{\mu}_{n_s}) \mid \dots \mid \mathbf{u}_h^{N_t}(\boldsymbol{\mu}_{n_s})\right],$$

where each $\mathbf{u}_h^n(\boldsymbol{\mu}_\ell)$ corresponds to the FOM solution for a fixed parameter $\boldsymbol{\mu}_\ell$ at time t^n, for $n = 1, \dots, N_t$. The reduced basis $\mathbf{V} \in \mathbb{R}^{N_h \times N}$ for the solution-space is obtained by performing POD on $\mathbf{S}_u$, so that a global basis is used to capture parameter and space-time variability simultaneously. We point out that different strategies are also possible, as done, e.g., in [Pfa+20].

To conclude, the RB approximation to $\mathbf{u}_h^n(\boldsymbol{\mu})$, for $n = 1, \dots, N_t$, is given by the resulting linear combination $\mathbf{V}\mathbf{u}_N^n(\boldsymbol{\mu})$, where $\mathbf{u}_N^n(\boldsymbol{\mu}) \in \mathbb{R}^N$ is the solution to the reduced nonlinear system

$$\mathbf{V}^T\mathbf{R}(\mathbf{V}\mathbf{u}_N^n(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}) = \mathbf{0},$$

arising from a Galerkin projection of the high-fidelity problem onto the subspace spanned by the columns of the basis $\mathbf{V}$. By employing Newton method, we need to solve – online – the following sequence of problems: given $\boldsymbol{\mu} \in \mathcal{P}$ and, for $n = 1, \dots, N_t$, the initial guess $\mathbf{u}_N^{n,(0)}(\boldsymbol{\mu}) = \mathbf{u}_N^{n-1}(\boldsymbol{\mu})$, find $\delta\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}) \in \mathbb{R}^N$ such that, for $k \geq 0$,

$$\begin{cases} \mathbf{V}^T\mathbf{J}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu})\mathbf{V}\delta\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}) = -\mathbf{V}^T\mathbf{R}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}), \\ \mathbf{u}_N^{n,(k+1)}(\boldsymbol{\mu}) = \mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}) + \delta\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}), \end{cases} \tag{2.7}$$

until $\|\mathbf{V}^T\mathbf{R}(\mathbf{V}\mathbf{u}_N^{n,(k+1)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu})\|_2 / \|\mathbf{V}^T\mathbf{R}(\mathbf{V}\mathbf{u}_N^{n,(0)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu})\|_2 < \varepsilon_{Nwt}$, where $\varepsilon_{Nwt} > 0$ is a chosen tolerance.

Crucial in the efficacy of the RB method is the decoupling of the offline and online stages, ensuring that the online computations are independent of the FOM dimension. However, since both the residual vector $\mathbf{R}$ and the Jacobian matrix $\mathbf{J}$ depend on the solution at the previous iteration $\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu})$, they need to be assembled at each Newton step. This means that, in order to set up the reduced system $(2.7)_1$, we need to assemble the high-dimensional arrays before projecting them onto the reduced space spanned by the columns of $\mathbf{V}$, entailing a computational complexity that still depends on (suitable powers of) N_h.

To overcome this issue, a further level of approximation, know as hyper-reduction, must be introduced, thus pursuing an *approximate-then-reduce* strategy.

2.2 System approximation: the (discrete) empirical interpolation method

In the case of PDEs featuring nonaffine dependence on the parameters and/or nonlinear (high-order polynomial, or non-polynomial) dependence on the field variable, a further level of reduction must be introduced to guarantee the offline-online decoupling in the ROM construction [Far+20]. To recover the ROM efficiency, state-of-the-art methods, such as the empirical interpolation method (EIM) [Bar+04; Gre+07], the discrete empirical interpolation method (DEIM) [CS10], its variant matrix DEIM [WSH14; CTB15; NMA15], the missing point estimation [Ast+08] and the Gauss-Newton with approximated tensors [CBMF11], aim at recover an affine expansion of the nonlinear operators by computing only a few entries of the nonlinear terms.

Although originally developed in the context of nonaffine operators, DEIM represents a valid hyper-reduction technique also for nonlinear parameterized PDEs [DHO12; AZF12; TR13; HSZ14; RR16; GTS17; BMQ16; BMQ17], employing an interpolation scheme for the approximation of the nonlinear function. The key idea is to replace the nonlinear arrays in (2.7) with a collateral reduced-basis expansion, computed through a (hopefully, inexpensive) interpolation procedure. In the case of DEIM, the construction of the interpolation points, commonly referred to as *magic points*, is based on a greedy algorithm, while the (prior) construction of the reduced basis is obtained by performing POD (or randomized-SVD) on a set of proper snapshots; in the case of EIM, both tasks are performed at the same time, exploiting a greedy algorithm.

The DEIM algorithm for the approximation of a generic nonlinear function $\mathbf{f} : \mathcal{P} \to \mathbb{R}^{N_h}$, $\mathbf{f} = \mathbf{f}(\tau)$ (where $\tau = t$ and/or $\boldsymbol{\mu}$), as originally proposed in [CS10], is outlined as follows[1]:

1. construct a set of snapshots obtained by sampling $\mathbf{f}(\tau)$ at random values $\tau_1, \dots, \tau_{n_s}$ and apply POD to extract a basis from these snapshots, i.e.

$$\mathbb{R}^{N_h \times m} \ni \boldsymbol{\Phi}_{\mathcal{F}} = \text{POD}([\mathbf{f}(\tau_1), \dots, \mathbf{f}(\tau_{n_s})], \varepsilon_{POD}),$$

   where $\varepsilon_{POD} > 0$ is a given tolerance such that (2.6) holds;

2. iteratively select $m \ll N_h$ indices $\mathcal{I} \subset \{1, \dots, N_h\}$, corresponding to a subset rows of $\boldsymbol{\Phi}_{\mathcal{F}}$, using a greedy procedure, which minimizes the interpolation error over the snapshots set;

3. given $\tau \notin \{\tau_1, \dots, \tau_{n_s}\}$, impose the interpolation conditions at the selected entries $\mathcal{I}$

$$\boldsymbol{\Phi}_{\mathcal{F}|_{\mathcal{I}}} \mathbf{c}(\tau) = \mathbf{f}(\tau)_{|_{\mathcal{I}}}, \tag{2.8}$$

   stating that the DEIM approximation $\boldsymbol{\Phi}_{\mathcal{F}}\mathbf{c}(\tau)$ and the original function $\mathbf{f}(\tau)$ must match at the magic points. Here $\boldsymbol{\Phi}_{\mathcal{F}|_{\mathcal{I}}} \in \mathbb{R}^{m \times m}$ is the matrix formed by the $\mathcal{I}$ rows of $\boldsymbol{\Phi}_{\mathcal{F}}$; as a result, we obtain

$$\mathbf{f}(\tau) \approx \mathbf{f}_m(\tau) := \boldsymbol{\Phi}_{\mathcal{F}} \boldsymbol{\Phi}_{\mathcal{F}}^{-1}{}_{|_{\mathcal{I}}} \mathbf{f}(\tau)_{|_{\mathcal{I}}}.$$

Algorithm 4 illustrates the greedy procedure to determine the DEIM interpolation points. The first magic point is the dof, in the physical domain, corresponding to the largest absolute

[1] for the sake of simplicity, we directly consider the vector representation of a function, assuming that all the quantities introduced in this procedure have been discretized on a FE mesh.

value of the first basis vector of $\boldsymbol{\Phi}_{\mathcal{F}}$. For $k \in \{2, \dots, m\}$, the k-th magic point added to the set is the dof corresponding to the largest absolute value of the difference between the current basis vector $\boldsymbol{\phi}_k$ and its current approximation $\boldsymbol{\Phi}_{\mathcal{F}}(:, 1\colon k-1)\boldsymbol{\psi}$, based on the $k-1$ computed interpolation points. In this way the magic points are hierarchical and non-repeated.

Algorithm 4 Discrete empirical interpolation method (DEIM)

INPUT: $\boldsymbol{\Phi}_{\mathcal{F}} = [\boldsymbol{\phi}_1, \dots, \boldsymbol{\phi}_m] \in \mathbb{R}^{N_h \times m}$
OUTPUT: $\mathcal{I} = \{i_1, \dots, i_m\}$

Find $i_1 = \arg\max_i \{|\boldsymbol{\phi}_1|_i\}_{i=1}^{N_h}$
Set $\mathcal{I} = \{i_1\}$
for $k = 2, \dots, m$ **do**
  Solve $\boldsymbol{\Phi}_{\mathcal{F}}(:, 1\colon k-1)_{|_{\mathcal{I}}}\boldsymbol{\psi} = \boldsymbol{\phi}_{k|_{\mathcal{I}}}$
  Compute $\mathbf{r} = \boldsymbol{\phi}_k - \boldsymbol{\Phi}_{\mathcal{F}}(:, 1\colon k-1)\boldsymbol{\psi}$
  Find $i_k = \arg\max_i \{|\mathbf{r}|_i\}_{i=1}^{N_h}$
  Set $\mathcal{I} \leftarrow \mathcal{I} \cup \{i_k\}$
end for

We remark that the interpolation condition (2.8) can be generalized to the case where more sample indices than basis function are considered, leading to a gappy POD reconstruction. The solution to the least-squares problem

$$\mathbf{c}(\tau) = \arg\min_{\mathbf{x} \in \mathbb{R}^m} \|\mathbf{f}(\tau)_{|_{\mathcal{I}}} - \boldsymbol{\Phi}_{\mathcal{F}|_{\mathcal{I}}}\mathbf{x}\|_2$$

would yield, in this case, $\mathbf{f}_m(\tau) = \boldsymbol{\Phi}_{\mathcal{F}}\boldsymbol{\Phi}^{\dagger}_{\mathcal{F}|_{\mathcal{I}}}\mathbf{f}(\tau)_{|_{\mathcal{I}}}$, where the Moore-Penrose inverse of a full column rank matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$ is defined as $\mathbf{A}^{\dagger} := (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$.

For the case at hand, the high-dimensional residual $\mathbf{R}$ is projected onto a reduced subspace of dimension $m < N_h$ spanned by a basis $\boldsymbol{\Phi}_{\mathcal{R}} \in \mathbb{R}^{N_h \times m}$

$$\mathbf{R}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}) \approx \boldsymbol{\Phi}_{\mathcal{R}}\mathbf{c}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}),$$

where $\mathbf{c} \in \mathbb{R}^m$ is the vector of the unknown amplitudes. The matrix $\boldsymbol{\Phi}_{\mathcal{R}}$ can be pre-computed offline by performing POD on a set of high-fidelity residuals collected when solving (2.7) for n'_s training input parameters (different from the one used for the RB basis construction),

$$\mathbf{S}_R = \left[\mathbf{R}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}_{\ell'}), t^n; \boldsymbol{\mu}_{\ell'})), k \geq 0\right]_{n=1:N_t}^{\ell'=1:n'_s}. \tag{2.9}$$

Remark 2.2
In addition to the set of snapshots (2.9), we can consider the FOM residuals collected when solving (1.13) during the RB-basis construction, i.e.

$$\begin{aligned}\mathbf{S}_R &= \mathbf{S}_R^{FOM} \cup \mathbf{S}_R^{ROM} \\ &= \left[\mathbf{R}(\mathbf{u}_h^{n,(k)}(\boldsymbol{\mu}_\ell), t^n; \boldsymbol{\mu}_\ell)), k \geq 0\right]_{n=1:N_t}^{\ell=1:n_s} \cup \left[\mathbf{R}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}_{\ell'}), t^n; \boldsymbol{\mu}_{\ell'})), k \geq 0\right]_{n=1:N_t}^{\ell'=1:n'_s}.\end{aligned}$$

Taking into account both FOM and ROM residuals entails no extra computational cost and usually improves the overall accuracy, assuming $n'_s > n_s$. On the contrary, collecting FOM

residuals only gave inaccurate results for all the performed test cases, as shown, for the stationary case, in [BMQ17]. In fact, DEIM aims at approximating the nonlinear operators evaluated at the ROM solution, rather than at the FOM solution. Therefore, it follows that construction of the reduced space and system approximation must be performed sequentially.

For every new instance of the parameter, the $\boldsymbol{\mu}$-dependent coefficient vector $\mathbf{c}$ is efficiently evaluated online by collocating the approximation at the m components selected by a greedy procedure, that is,

$$\boldsymbol{\Phi}_{\mathcal{R}|_{\mathcal{I}}}\mathbf{c}(\mathbf{V}\mathbf{u}_N^{n,(k)}, t^n; \boldsymbol{\mu}) = \mathbf{R}(\mathbf{V}\mathbf{u}_N^{n,(k)}, t^n; \boldsymbol{\mu})_{|_{\mathcal{I}}},$$

where $\boldsymbol{\Phi}_{\mathcal{R}|_{\mathcal{I}}}$ and $\mathbf{R}(\cdot)_{|_{\mathcal{I}}}$ are the restrictions of $\boldsymbol{\Phi}_{\mathcal{R}}$ and $\mathbf{R}(\cdot)$ to the subset of indices $\mathcal{I}$, respectively. We thus define the hyper-reduced residual vector approximating $\mathbf{V}^T\mathbf{R}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu})$ as

$$\mathbf{R}_{N,m}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}) := \mathbf{V}^T\boldsymbol{\Phi}_{\mathcal{R}}(\boldsymbol{\Phi}_{\mathcal{R}}{}^{-1})_{|_{\mathcal{I}}}\mathbf{R}(\mathbf{V}\mathbf{u}_N^{n,(k)}, t^n; \boldsymbol{\mu})_{|_{\mathcal{I}}}.$$

During the online phase we need to assemble the $\boldsymbol{\mu}$-dependent quantities $\mathbf{R}(\mathbf{V}\mathbf{u}_N^{n,(k)}, t^n; \boldsymbol{\mu})_{|_{\mathcal{I}}}$ only, which are vectors of (possibly small) dimension m. All other quantities are constant (in fact, $\boldsymbol{\Phi}_{\mathcal{R}}$ does not depend on $t > 0$, nor on $\boldsymbol{\mu} \in \mathcal{P}$) and can be pre-computed and stored offline. Finally, the Jacobian approximation to $\mathbf{V}^T\mathbf{J}(\mathbf{V}\mathbf{u}_N^{n,(k)}, t^n; \boldsymbol{\mu})\mathbf{V}$ can be computed as the derivative of $\mathbf{R}_{N,m}(\mathbf{V}\mathbf{u}_N^{n,(k)}, t^n; \boldsymbol{\mu})$ with respect to the reduced displacement, that is,

$$\begin{aligned}
\mathbf{J}_{N,m}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}) :=& \frac{\partial \mathbf{R}_{N,m}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu})}{\partial \mathbf{u}_N} \\
=& \mathbf{V}^T\boldsymbol{\Phi}_{\mathcal{R}}(\boldsymbol{\Phi}_{\mathcal{R}}{}^{-1})_{|_{\mathcal{I}}} \frac{\partial \mathbf{R}(\mathbf{V}\mathbf{u}_N^{n,(k)}, t^n; \boldsymbol{\mu})}{\partial \mathbf{u}_N}|_{\mathcal{I}} \\
=& \mathbf{V}^T\boldsymbol{\Phi}_{\mathcal{R}}(\boldsymbol{\Phi}_{\mathcal{R}}{}^{-1})_{|_{\mathcal{I}}}\mathbf{J}(\mathbf{V}\mathbf{u}_N^{n,(k)}, t^n; \boldsymbol{\mu})_{|_{\mathcal{I}}}\mathbf{V},
\end{aligned}$$

or by relying on the MDEIM algorithm, as done in [BMQ16]. However, since we employ automatic differentiation to get (an approximation of) the Jacobian matrices, we adopt the former approximation. As before, the only quantity that must be computed online is the restriction of the Jacobian matrix to the rows corresponding to the magic points, i.e. $\mathbf{J}(\mathbf{V}\mathbf{u}_N^{n,(k)}, t^n; \boldsymbol{\mu})_{|_{\mathcal{I}}} \in \mathbb{R}^{m \times N_h}$. Note that employing DEIM can be regarded as the use of an exact Newton method on the reduced problem $\mathbf{R}_{N,m}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}) = \mathbf{0}$, such that the k-th Newton iteration for its solution reads

$$\begin{cases}
\mathbf{J}_{N,m}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu})\delta\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}) = -\mathbf{R}_{N,m}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}), \\
\mathbf{u}_N^{n,(k+1)}(\boldsymbol{\mu}) = \mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}) + \delta\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}).
\end{cases} \tag{2.10}$$

DEIM eliminates any full order evaluation, highly decreasing the computational effort, provided its dimension is not too large [BGW15].

Remark 2.3

The m points selected by the DEIM algorithm correspond to a subset of nodes of the computational mesh, which, together with the neighboring nodes (i.e. those sharing the same cell), form the so-called *reduced mesh*; an instance of reduced mesh is reported in Figure 2.1. Since the entries of any FE-vector are associated with the dofs of the problem, $\mathbf{R}_{N,m}$ and $\mathbf{J}_{N,m}$ can be computed by integrating the corresponding FOM residual and Jacobian only on the quadrature points belonging to the reduced mesh, respectively.

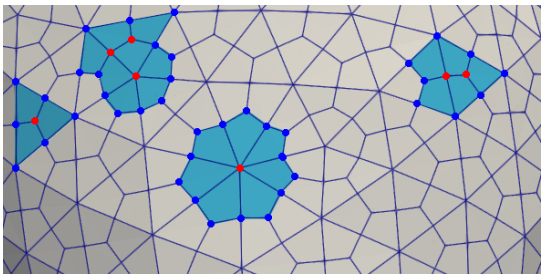


Figure 2.1: Sketch of a reduced mesh for an hexahedral computational grid in a two-dimensional case. The red dots represent the points selected by the DEIM algorithm.

Algorithm 5 POD-Galerkin-DEIM for nonlinear time-dependent problems

Offline stage

INPUT: $\boldsymbol{\mu}_\ell$, for $\ell = 1, \dots, n_s$, and $\boldsymbol{\mu}_{\ell'}$, for $\ell' = 1, \dots, n'_s$

OUTPUT: $\mathbf{V} \in \mathbb{R}^{N_h \times N}$, $\boldsymbol{\Phi}_{\mathcal{R}} \in \mathbb{R}^{N_h \times m}$, $\mathcal{I} = \{i_1, \dots, i_m\}$

for $\ell = 1, \dots, n_s$ **do**
  for $n = 1, \dots, N_t$ **do**
    for $k \geq 0$ **until convergence do**
      Assemble and solve problem (1.13)
      Collect $\mathbf{S}_u \leftarrow \mathbf{S}_u \cup \left[\mathbf{u}_h^{n,(k)}(\boldsymbol{\mu}_\ell)\right]$ column-wise
      Collect $\mathbf{S}_R \leftarrow \mathbf{S}_R \cup \left[\mathbf{R}(\mathbf{u}_h^{n,(k)}(\boldsymbol{\mu}_{\ell'}), t^n; \boldsymbol{\mu}_{\ell'})\right]$ column-wise (see Remark 2.2)
Construct $\mathbf{V} = \text{POD}(\mathbf{S}_u, \varepsilon_{POD})$ (see Algorithm 2)
for $\ell' = 1, \dots, n'_s$ **do**
  for $n = 1, \dots, N_t$ **do**
    for $k \geq 0$ **until convergence do**
      Assemble and solve reduced problem (2.7)
      Collect $\mathbf{S}_R \leftarrow \mathbf{S}_R \cup \left[\mathbf{R}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}_{\ell'}), t^n; \boldsymbol{\mu}_{\ell'})\right]$ column-wise
Construct $(\boldsymbol{\Phi}_{\mathcal{R}}, \mathcal{I}) = \text{DEIM}(\mathbf{S}_R, \varepsilon_{DEIM})$ (see Algorithm 4)

Online stage

INPUT: $\boldsymbol{\mu} \in \mathcal{P}$

OUTPUT: $\mathbf{V}\mathbf{u}_N^n(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$, for $n = 1, \dots, N_t$

for $n = 0, \dots, N_t - 1$ **do**
  for $k \geq 0$ **until convergence do**
    Assemble and solve hyper-reduced problem (2.10)
Recover $\mathbf{V}\mathbf{u}_N^n(\boldsymbol{\mu})$, for $n = 1, \dots, N_t$

Remark 2.4

We recall that both the FOM and the ROM arrays, such as the solution and the residual vectors, are column vectors whose elements are the values of the associated quantities evaluated on the dofs of the physical mesh. Therefore, when assembling the snapshots matrices in order to compute the RB and the DEIM basis, the corresponding arrays are stacked column-wise.

2.3 Numerical results

In this section we investigate the performances of the POD-Galerkin-DEIM approach on different applications related to the parameterized nonlinear time-dependent mechanics problem (1.2), namely:

(i) a series of structural tests on a rectangular beam, with different loading conditions and a simple nonlinear constitutive law;

(ii) two test cases on an idealized left ventricle geometry, simulating cardiac relaxation and contraction, respectively. The steady-state versions of these test cases have been introduced in [Lan+15] as benchmarks for the validation of cardiac mechanics software;

(iii) an idealized full cycle of a patient-specific left ventricle, where both pressure and active stress are imposed.

In all these scenarios, we neglect the body forces $\mathbf{b}_0(\boldsymbol{\mu}) = \mathbf{0}$ and consider zero initial conditions $\mathbf{u}_0(\boldsymbol{\mu}) = \dot{\mathbf{u}}_0(\boldsymbol{\mu}) = \mathbf{0}$. Regarding the Robin boundary, we always assume $\alpha = \beta = 0$, so that we actually impose homogeneous Neumann conditions. Finally, the traction vector $\bar{\mathbf{T}}$ is given by

$$\bar{\mathbf{T}}(\mathbf{X}, t, \mathbf{N}; \boldsymbol{\mu}) = -\mathbf{g}(t; \boldsymbol{\mu}) J \mathbf{F}^{-T} \mathbf{N},$$

where $\mathbf{g}(t; \boldsymbol{\mu})$ will be specified, according to the application at hand.

As a measure of the accuracy of the ROMs with respect to the FOM, for a given parameter instance, we consider time-averaged L^2-errors of the displacement vector, that are,

$$\epsilon_{abs}(\boldsymbol{\mu}) = \frac{1}{N_t} \sum_{n=1}^{N_t} \|\mathbf{u}_h(\cdot, t^n; \boldsymbol{\mu}) - \mathbf{V}\mathbf{u}_N(\cdot, t_n; \boldsymbol{\mu})\|_2,$$

$$\epsilon_{rel}(\boldsymbol{\mu}) = \frac{1}{N_t} \sum_{n=1}^{N_t} \frac{\|\mathbf{u}_h(\cdot, t^n; \boldsymbol{\mu}) - \mathbf{V}\mathbf{u}_N(\cdot, t_n; \boldsymbol{\mu})\|_2}{\|\mathbf{u}_h(\cdot, t^n; \boldsymbol{\mu})\|_2}.$$

The CPU time ratio, i.e. the ratio between FOM and ROM computational times, is used to measure the ROMs efficiency, since it represents the speed-up of the ROMs with respect to the FOM. All the computations have been performed on a PC desktop computer with 3.70GHz Intel Core i5-9600K CPU and 16GB RAM.

2.3.1 Deformation of a clamped rectangular beam

The first series of test cases represents a typical structural mechanics problem, with reference geometry $\bar{\Omega}_0 = [0, 10^{-2}] \times [0, 10^{-3}] \times [0, 10^{-3}]$ m, see Figure 2.2. We consider a nearly-incompressible neo-Hookean material characterized by the following strain density energy function

$$\mathcal{W}(\mathbf{F}) = \frac{G}{2}(\mathcal{I}_1 - 3) + \frac{K}{4}((J-1)^2 + \ln^2(J)),$$

where $G > 0$ is the shear modulus, $\mathcal{I}_1 = J^{-\frac{2}{3}} \det(\mathbf{C})$ and the latter term is needed to enforce incompressibility, being the bulk modulus $K > 0$ the penalization factor. This choice leads to the following first Piola-Kirchhoff stress tensor, characterized by a non-polynomial nonlinearity,

$$\mathbf{P}(\mathbf{F}) = G J^{-\frac{2}{3}} \left(\mathbf{F} - \frac{1}{3} \mathcal{I}_1 \mathbf{F}^T \right) + \frac{K}{2} J \left(J - 1 + \frac{1}{J} \ln(J) \right) \mathbf{F}^T.$$

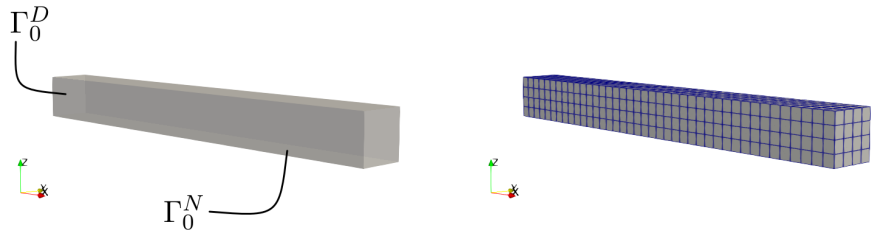


Figure 2.2: Rectangular beam geometry (left) and computational grid (right).

Pressure boundary conditions, changing with the deformed surface orientation, are applied to the entire bottom face $z = 0$ m (i.e. Γ_0^N), whereas Dirichlet boundary conditions are imposed on the left face $x = 0$ m (i.e. Γ_0^D), that is, the beam is clamped at the left-hand side. For all other faces, we consider homogeneous Neumann conditions (i.e. Γ_0^R with $\alpha = \beta = 0$). As possible functions for the external load $\mathbf{g}(t; \boldsymbol{\mu})$, we choose

1. a linear function $\mathbf{g}(t; \boldsymbol{\mu}) = \widetilde{p}\ t/T$;
2. a triangular or hat function $\mathbf{g}(t; \boldsymbol{\mu}) = \widetilde{p}\ \left(2t\ \chi(t)_{(0,\frac{T}{2}]} + 2(T-t)\ \chi(t)_{(\frac{T}{2},T]}\right)$;
3. a step function $\mathbf{g}(t; \boldsymbol{\mu}) = \widetilde{p}\ \chi(t)_{(0,\frac{T}{3}]}$, such that the presence of the inertial term is not negligible.

Here, $\widetilde{p} > 0$ is a parameter controlling the maximum load. The FOM is built on a hexahedral mesh with 640 elements and 1025 vertices, resulting in a high-fidelity dimension $N_h = 3075$ (since $\mathbb{Q}_1$-FE are employed). The resulting computational mesh in the reference configuration is reported in Figure 2.2.

Test case 1: linear function for the pressure load

First, we consider the parameterized linear function

$$\mathbf{g}(t; \boldsymbol{\mu}) = \widetilde{p}\ t/T$$

for the pressure load, describing a situation in which a structure is progressively loaded. We choose a time interval $t \in (0, 0.25)$ s and employ a uniform time step $\Delta t = 5 \cdot 10^{-3}$ s for the time discretization scheme, resulting in a total number of 50 time iterations. As parameters, we consider:

- the shear modulus $G \in [0.5 \cdot 10^4, 1.5 \cdot 10^4]$ Pa,
- the bulk modulus $K \in [2.5 \cdot 10^4, 7.5 \cdot 10^4]$ Pa, and
- the external load parameter $\widetilde{p} \in [2, 6]$ Pa.

In the following, we analyze the accuracy and the efficiency of the ROM without hyper-reduction with respect to the POD tolerance ε_{POD}. Then, for a fixed basis $\mathbf{V} \in \mathbb{R}^{N_h \times N}$, the performances of POD-Galerkin-DEIM are investigated, and the CPU times required for the assembling and solving of the reduced system are computed.

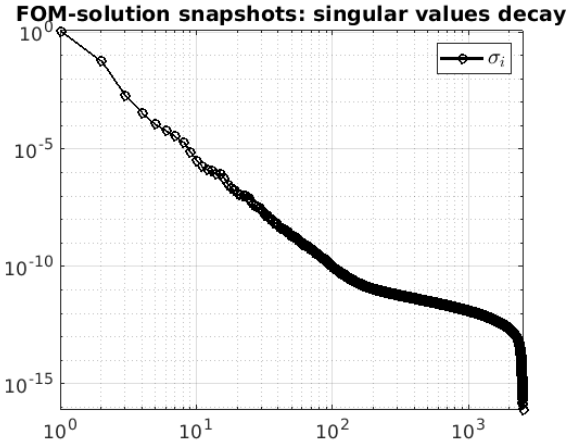


Figure 2.3: Test case 1. Decay of the singular values of the FOM solution snapshots matrix.

Let us consider a training set of $n_s = 50$ points generated from the three-dimensional parameter space $\mathcal{P}$ through latin hypercube sampling (LHS) and compute the reduced basis $\mathbf{V} \in \mathbb{R}^{N_h \times N}$ using the POD method with tolerance

$$\varepsilon_{POD} \in \{10^{-3}, 5 \cdot 10^{-4}, 10^{-4}, 5 \cdot 10^{-5}, 10^{-5}, 5 \cdot 10^{-6}, 10^{-6}\}.$$

The corresponding reduced dimensions are $N = 3$, 4, 5, 6, 8, 9 and 15, respectively. In Figure 2.3 we show the singular values of the snapshot matrix related to the FOM displacement $\mathbf{u}_h$, where a rapid decay of the plotted quantity means that a small number of RB functions are needed to correctly approximate the high-fidelity solution. The average relative error ϵ_{rel} computed over a testing set of 50 randomly chosen parameters, different from the ones used to compute the solution snapshots, is reported in Figure 2.4, together with the CPU time ratio. The approximation error decreases up to two orders of magnitude when reducing the POD tolerance

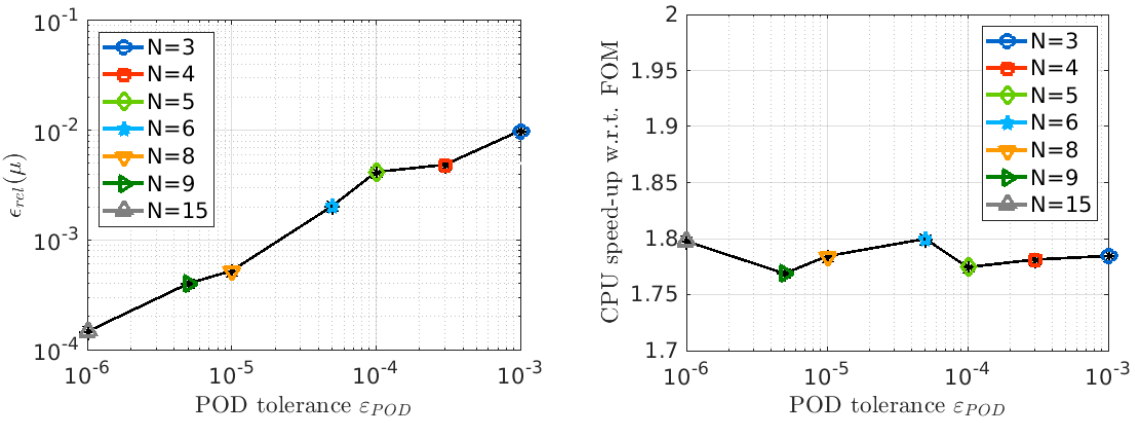


Figure 2.4: Test case 1. Average over 50 testing parameters of relative error ϵ_{rel} (left) and speed-up (right) of POD-Galerkin ROMs.

ε_{POD} from 10^{-3} to 10^{-6}, corresponding to an increase of the RB dimension from $N = 3$ to $N = 15$. Despite being the RB space low-dimensional, the computational speed-up achieved by the reduced model is negligible. This is due to the fact that the ROMs still depend on the FOM dimension N_h during the online stage. For this reason, we need to rely on suitable hyper-reduction techniques.

For the construction of the DEIM basis $\mathbf{\Phi}_{\mathcal{R}}$ for the approximation of the residual, we need first to compute snapshots from the ROM solutions for given parameter values and time instants, and then evaluate the residual vectors. To this goal, we choose a POD-Galerkin ROM with dimension $N = 4$, being it a good balance between accuracy and computational effort for the test case at hand, and perform ROM simulations (see Algorithm 5) for a given set of $n'_s = 200$ parameter samples to collect residual data. We report in Figure 2.5 the decay of the singular values of $\mathbf{S}_R$ related to the residual snapshots $\mathbf{R}(\mathbf{Vu}_N^{n,(k)}(\boldsymbol{\mu}_{\ell'}), t^n; \boldsymbol{\mu}_{\ell'})$.

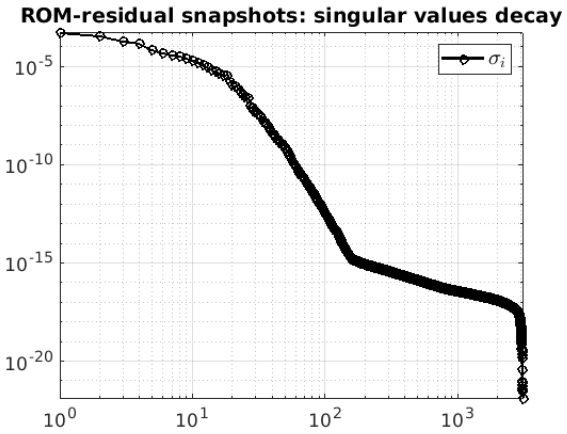


Figure 2.5: Test case 1. Decay of the singular values of the ROM (N=4) residual snapshots matrix.

In order to investigate the impact of hyper-reduction onto the ROM solution reconstruction error, we compute the DEIM basis $\mathbf{\Phi}_{\mathcal{R}}$ for the approximation of the residual using the POD method with different tolerances, that are

$$\varepsilon_{DEIM} \in \{10^{-3}, 5 \cdot 10^{-4}, 10^{-4}, 5 \cdot 10^{-5}, 10^{-5}, 5 \cdot 10^{-6}, 10^{-6}\},$$

corresponding to a number of interpolation points equal to $m = 22$, 25, 30, 33, 39, 43, 51, respectively. Larger POD tolerances were not sufficient to ensure the convergence of Newton method for all considered testing parameters. The average relative error ϵ_{rel} is evaluated over a testing set of $n'_s = 50$ parameters and plotted in Figure 2.6. In Figures 2.7 and 2.8 we report the high-fidelity, the reduced displacements and their difference, i.e. point-wise displacement $|\mathbf{u}_h^n(\boldsymbol{\mu}) - \mathbf{Vu}_N^n(\boldsymbol{\mu})|$, for 2 different values of the parameter $\boldsymbol{\mu} = [G, K, \tilde{p}]$. The reduced solution has been computed using a POD-Galerkin-DEIM hyper-reduced order model with $N = 4$ and $m = 22$. In order to compute the high-fidelity solutions, 26 s are required in average, while a POD-Galerkin-DEIM ROM requires only 2.4 s, thus yielding a speed-up of $\times 11$ compared to the FOM.

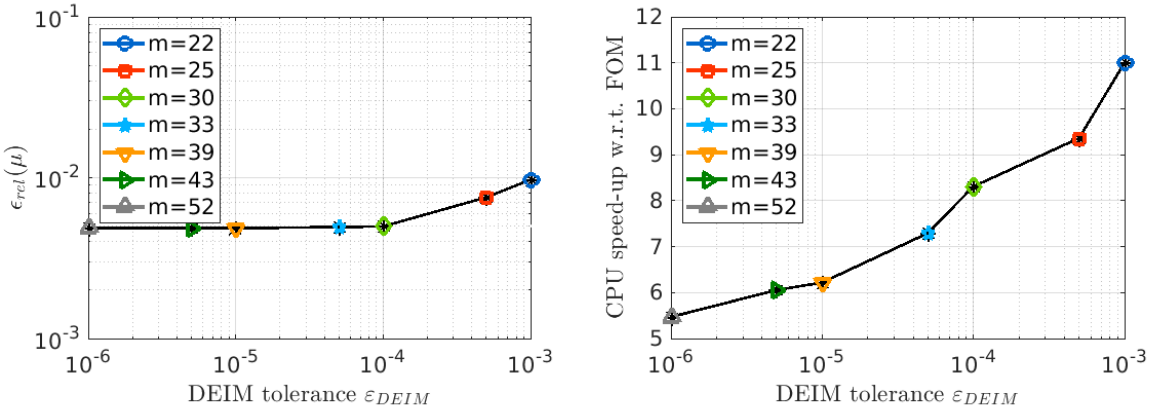


Figure 2.6: Test case 1. Average over 50 testing parameters of relative error ϵ_{rel} (left) and average speed-up (right) of POD-Galerkin-DEIM with $N = 4$.

Data related to the performances of the POD-Galerkin-DEIM method for $N = 4$ and different values of m are shown in Table 2.1. The number of elements of the reduced mesh represents a small percentage of the one forming the original grid, so that the cost related to the residual assembling is remarkably alleviated. Relying on the AD tool allows to drastically reduce the computational time required for the assembling of the Jacobian matrices.

POD tolerance ε_{DEIM}	$5 \cdot 10^{-4}$	$5 \cdot 10^{-5}$	$5 \cdot 10^{-6}$
DEIM interpolation dofs m	25	33	43
Reduced mesh elements (total: 640)	86	115	168
Online CPU time	2.8 s	3.6 s	4.3 s
◦ system construction [∗]	78%	83%	88%
◦ system solution	0.16%	0.13%	0.09%
[∗] System construction for each Newton iteration	0.02 s	0.02 s	0.03 s
◦ residual assembling	89%	87%	88%
◦ Jacobian computing through AD	0.6%	0.4%	0.5%
Computational speed-up	×9.4	×7.3	×6.0
Time-averaged $L^2(\Omega_0)$-absolute error	$3 \cdot 10^{-5}$	$2 \cdot 10^{-5}$	$2 \cdot 10^{-5}$
Time-averaged $L^2(\Omega_0)$-relative error	$8 \cdot 10^{-3}$	$5 \cdot 10^{-3}$	$5 \cdot 10^{-3}$

Table 2.1: Test case 1. Computational data related to POD-Galerkin-DEIM with $N = 4$ and different values of m.

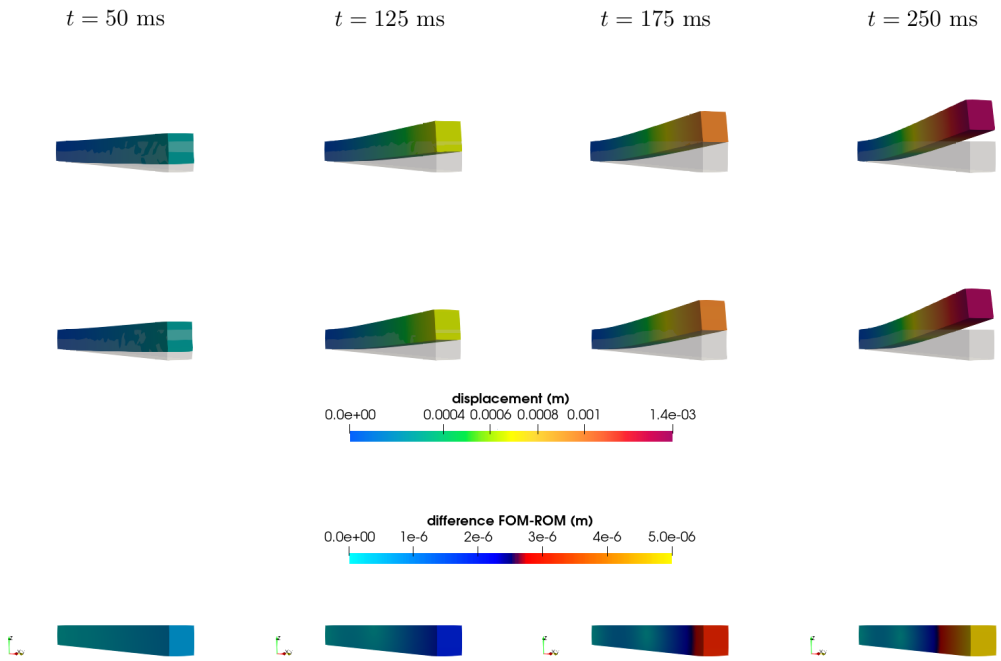


Figure 2.7: Test case 1. FOM (top) and POD-Galerkin-DEIM (middle) displacements at different times for $\boldsymbol{\mu} = [1.3225 \cdot 10^4 \text{ Pa}, 3.9875 \cdot 10^4 \text{ Pa}, 3.43 \text{ Pa}]$ and difference between them (bottom).

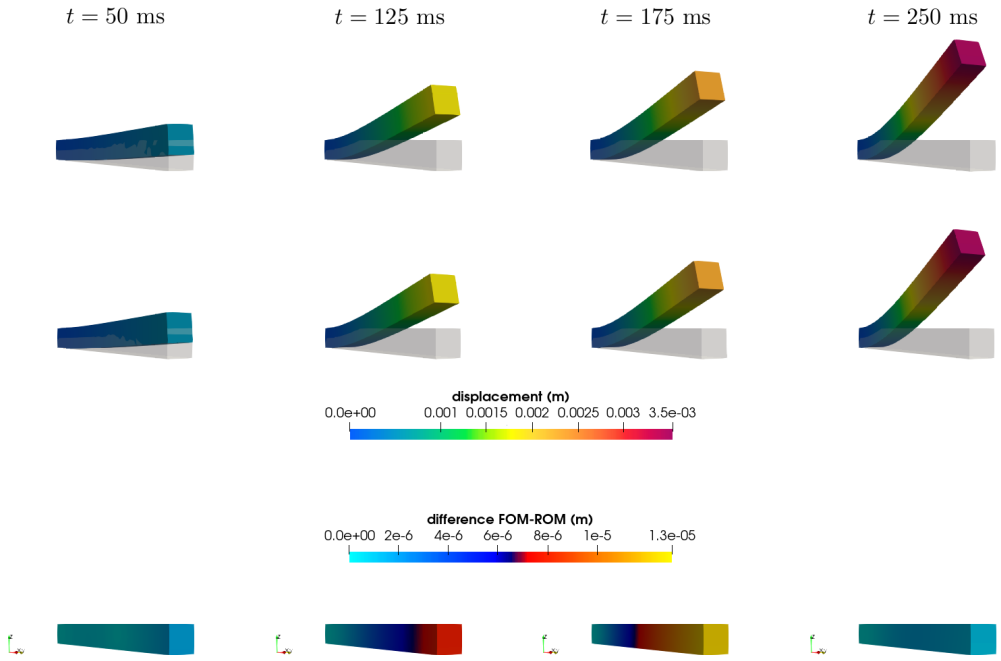


Figure 2.8: Test case 1. FOM (top) and POD-Galerkin-DEIM (middle) displacements at different times for $\boldsymbol{\mu} = [0.6625 \cdot 10^4 \text{ Pa}, 5.8625 \cdot 10^4 \text{ Pa}, 4.89 \text{ Pa}]$ and difference between them (bottom).

The final accuracy of the POD-Galerkin-DEIM method for $N = 4$ equals that of the ROM without hyper-reduction for all ε_{DEIM} considered, $\epsilon_{rel} \approx 10^{-2}$, meaning that the projection error dominates over the nonlinear operators approximation error. In order to increase the accuracy of the reduced solution, we need to consider higher values of N. In Figure 2.9 we report the average relative error and the CPU time ratio computed for $N = 8$. In this case we are able to improve the accuracy of the reduced solution of more than one order of magnitude by increasing the residual basis dimension m. Of course, slower POD-Galerkin-DEIM approximations are obtained, since we need to assemble a higher number of entries of the nonlinear terms.

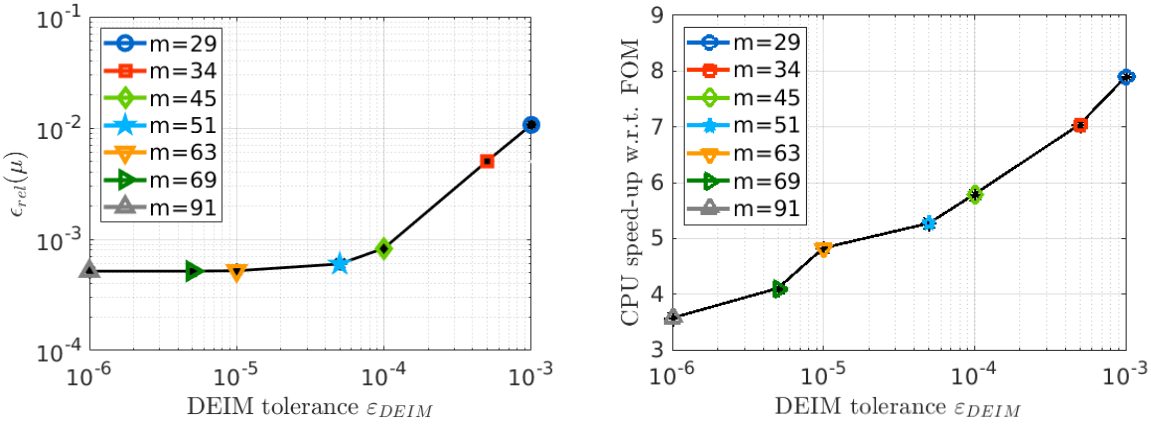


Figure 2.9: Test case 1. Average over 50 testing parameters of relative error ϵ_{rel} (left) and average speed-up (right) of POD-Galerkin-DEIM with $N = 8$.

Test case 2: hat function for the pressure load

Let us now consider a piecewise linear pressure load given by the following hat function

$$\mathbf{g}(t;\boldsymbol{\mu}) = \widetilde{p}\ \left(2t\ \chi(t)_{(0,\frac{T}{2}]} + 2(T-t)\ \chi(t)_{(\frac{T}{2},T]}\right),$$

describing the case in which a structure is increasingly loaded until a maximum pressure is reached, and then linearly unloaded in order to recover the initial resting state. For the case at hand, we choose $t \in (0, 0.35)$ s and $\Delta t = 5 \cdot 10^{-3}$ s, resulting in a total number of 70 time steps. As parameter, we consider the external load parameter $\widetilde{p} \in [2, 12]$ Pa. The shear modulus G and the bulk modulus K are fixed to the values 10^4 Pa and $5 \cdot 10^4$ Pa, respectively. Let us consider a training set of $n_s = 50$ points generated from the one-dimensional parameter space $\mathcal{P} = [2, 12]$ Pa through LHS and build the reduced space using $\mathbf{V} \in \mathbb{R}^{N_h \times N}$ with $N = 4$, corresponding to $\varepsilon_{POD} = 10^{-4}$.

The singular values of the solution snapshots matrix $\mathbf{S}_u$ and those of the matrix $\mathbf{S}_R$ related to the residual vectors $\mathbf{R}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}_{\ell'}), t^n; \boldsymbol{\mu}_{\ell'})$ computed for $n_s' = 300$ randomly chosen parameters are reported in Figure 2.10.

In this case the DEIM basis $\boldsymbol{\Phi}_{\mathcal{R}}$ for the approximation of the residual is computed using the POD method with tolerance

$$\varepsilon_{DEIM} \in \{10^{-3}, 5 \cdot 10^{-4}, 10^{-4}, 5 \cdot 10^{-5}, 10^{-5}, 5 \cdot 10^{-6}, 10^{-6}\},$$

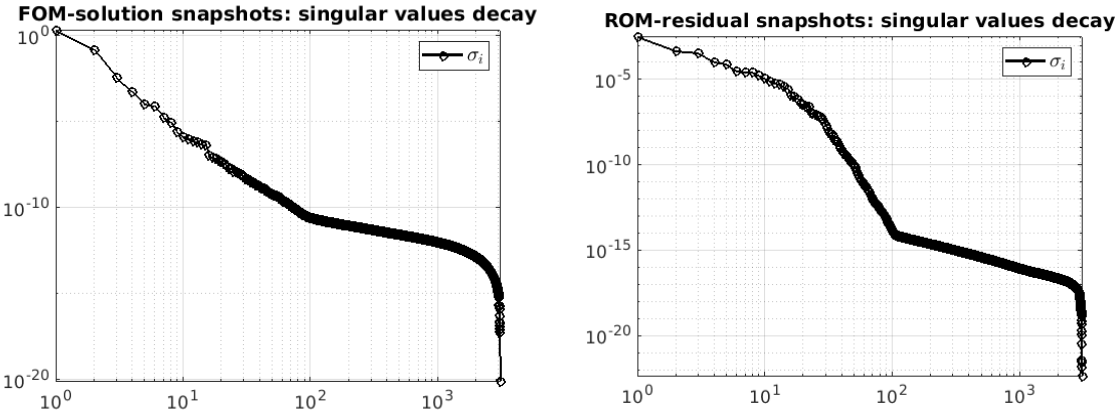


Figure 2.10: Test case 2. Decay of the singular values of the FOM solution (left) and the ROM (N=4) residual (right) snapshots matrices.

corresponding to a number of interpolation points $m = 14, 16, 22, 24, 29, 31, 37$, respectively. Tolerances ε_{DEIM} larger than the values reported above were not sufficient to ensure convergence of Newton method for all considered testing parameters. The average relative error ϵ_{rel} is about 10^{-2} when using $m = 14$ residual basis, and can be further reduced of one order of magnitude when increasing the DEIM dimension to $m = 29$, albeit highly decreasing the CPU time ratio, as shown in Figure 2.11. In Figure 2.12 we show the FOM solution, the POD-Galerkin-DEIM approximation and their difference for a randomly chosen value of $\widetilde{p}$, different from those used in the offline stage to train the ROM. This deformation corresponds to $N = 4$ and $m = 14$, showing that a good level of accuracy can be achieved also using a small number of basis functions.

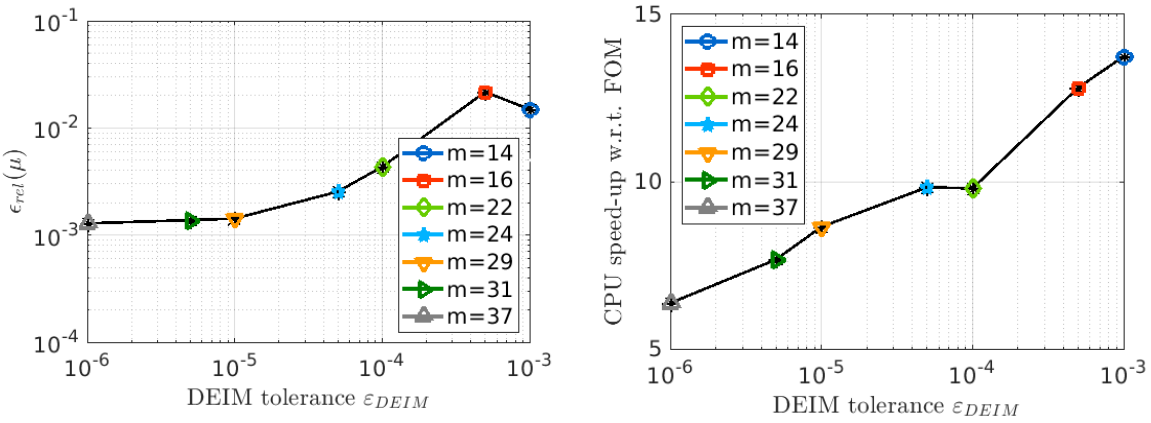


Figure 2.11: Test case 2. Average over 50 testing parameters of relative error ϵ_{rel} (left) and average speed-up (right) of POD-Galerkin-DEIM with $N = 4$.

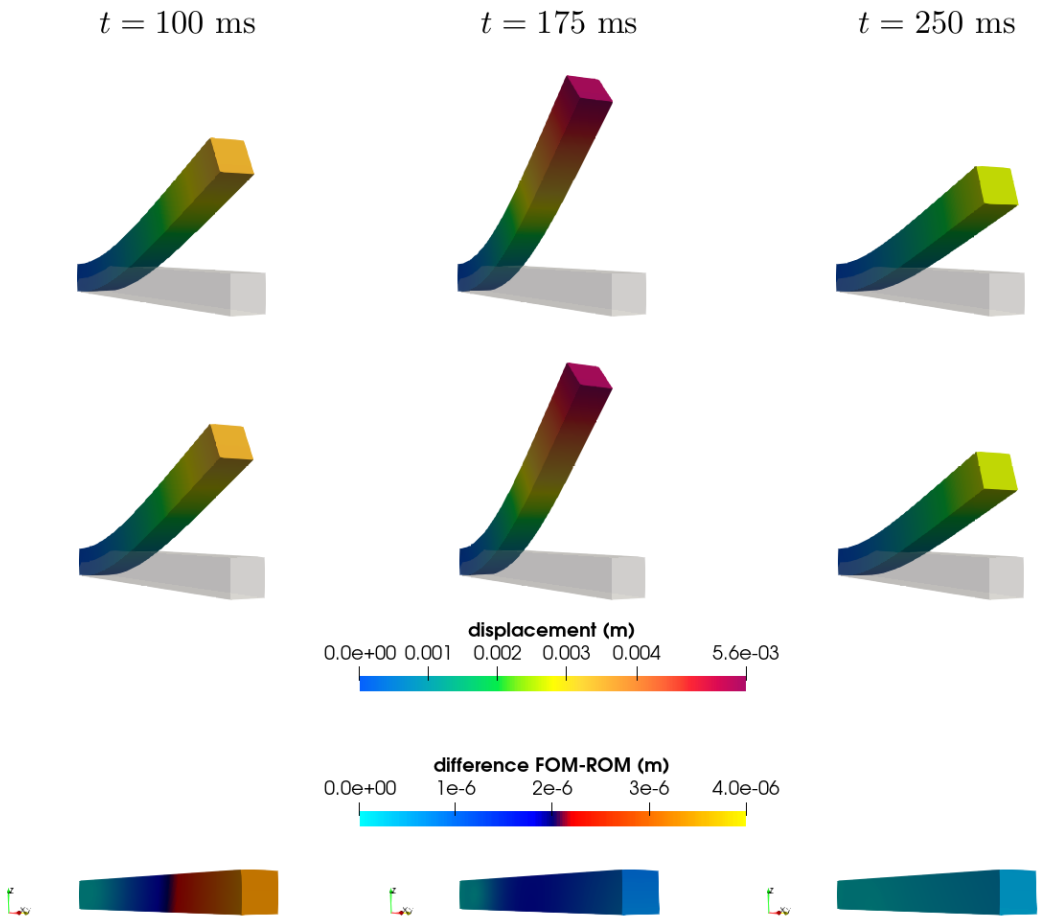


Figure 2.12: Test case 2. FOM (top) and POD-Galerkin-DEIM (middle) displacements at different times for $\boldsymbol{\mu} = [11.3875 \text{ Pa}]$ and difference between them (bottom).

Test case 3: step function for the pressure load

As last test case for the beam geometry, we consider a pressure load acting on the bottom surface area for only a third of the whole simulation time, that is

$$\mathbf{g}(t; \boldsymbol{\mu}) = \widetilde{p}\ \chi(t)_{(0, \frac{T}{3}]},$$

such that the resulting deformation features oscillations. This case is of particular interest in nonlinear elastodynamics, since the inertial term cannot be neglected, as it has a crucial impact on the deformation of the object. For the case at hand, we choose $t \in (0, 0.27)$ s and a uniform time step $\Delta t = 3.6 \cdot 10^{-3}$ s, resulting in a total number of 75 time iterations. For what concerns the input parameters, we vary the external load $\widetilde{p} \in [2, 12]$ Pa and consider $G = 10^4$ Pa and $K = 5 \cdot 10^4$ Pa fixed.

We build the reduced basis $\mathbf{V} \in \mathbb{R}^{N_h \times N}$ from a training set of $n_s = 50$ FOM solutions using $\varepsilon_{POD} = 10^{-3}$, obtaining a reduced dimension of $N = 4$, and perform POD-ROM simulations for a given set of $n_s' = 300$ parameter samples to collect residual data for the construction of POD-

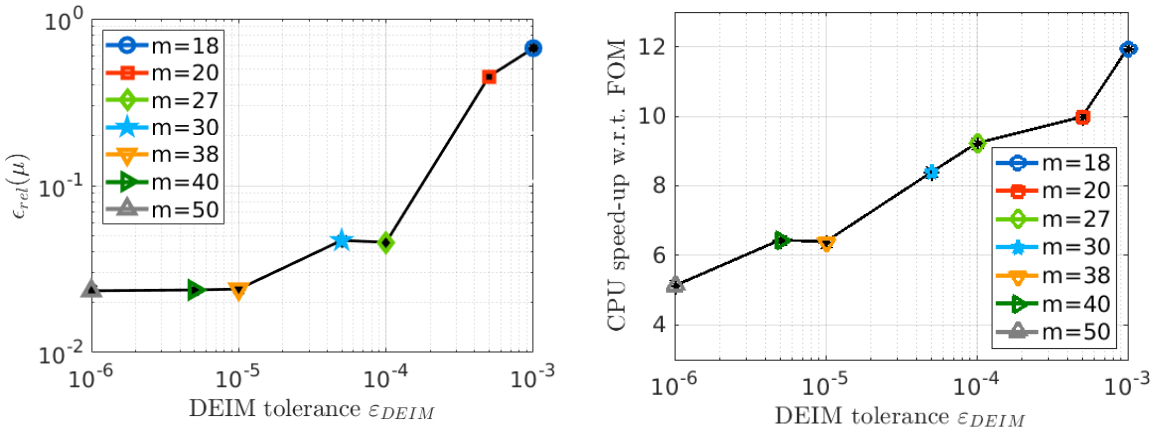


Figure 2.13: Test case 3. Average over 50 testing parameters of relative error ϵ_{rel} (left) and average speed-up (right) of POD-Galerkin-DEIM with $N = 4$.

Galerkin-DEIM model. Like previous test cases, the DEIM basis $\mathbf{\Phi}_{\mathcal{R}}$ for the approximation of the residual is computed by performing POD on $\mathbf{S}_R$ with tolerance

$$\varepsilon_{DEIM} \in \{10^{-3}, 5 \cdot 10^{-4}, 10^{-4}, 5 \cdot 10^{-5}, 10^{-5}, 5 \cdot 10^{-6}, 10^{-6}\},$$

where $\varepsilon_{DEIM} = 10^{-3}$ is the larger POD tolerance that allows to guarantee the convergence of the reduced Newton algorithm for all testing parameters. The corresponding number of interpolation points is $m = 18$, 20, 27, 30, 38, 40, 50, respectively. The results are shown in Figure 2.13.

Data related to the computational performances of POD-Galerkin-DEIM for three different dimensions of the residual basis are reported in Table 2.2. The POD-Galerkin-DEIM with $N = 4$ and $m = 27$ is able to accurately approximate the FOM solution, as shown in Figure 2.14, where we report the high-fidelity and the reduced solutions for a give pressure $\widetilde{p}$ at different time instances.

	$m = 18$	$m = 27$	$m = 38$
Computational speed-up	$\times 12$	$\times 9$	$\times 6$
Avg. CPU time	4 s	5 s	8 s
Time-avg. $L^2(\Omega_0)$-absolute error	$2.4 \cdot 10^{-3}$	$1.8 \cdot 10^{-4}$	$1.3 \cdot 10^{-4}$
Time-avg. $L^2(\Omega_0)$-relative error	$6.7 \cdot 10^{-1}$	$4.5 \cdot 10^{-1}$	$2.4 \cdot 10^{-2}$

Table 2.2: Test case 3. Computational data related to POD-Galerkin-DEIM with $N = 4$ and different values of m.

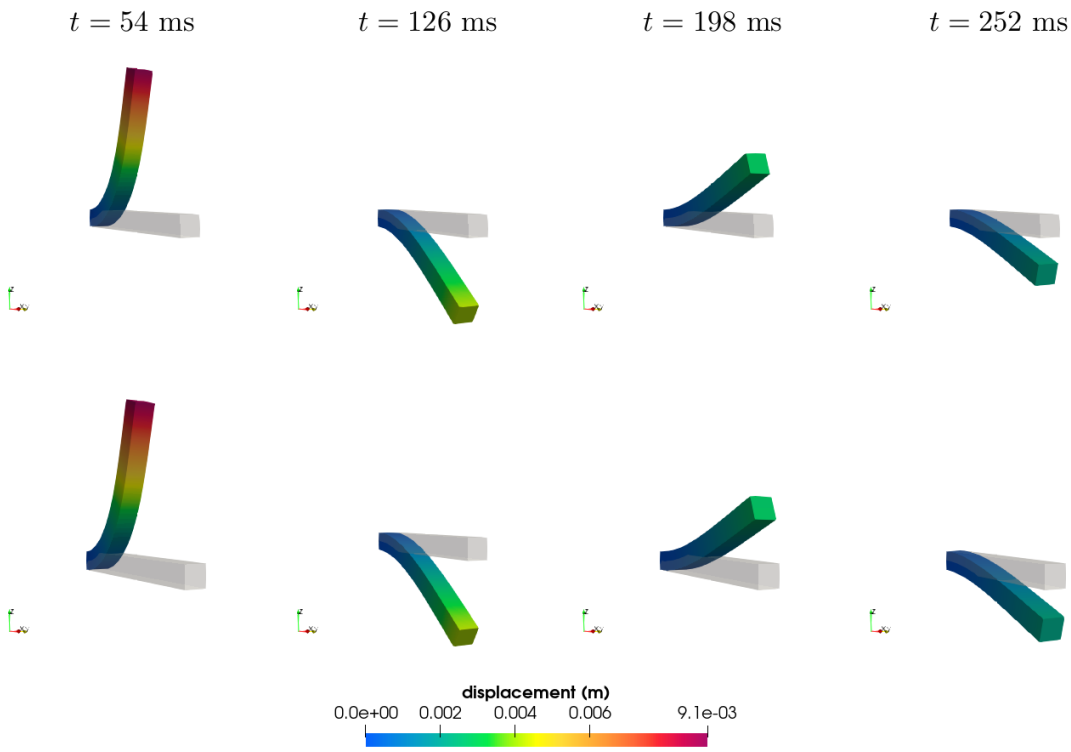


Figure 2.14: Test case 3. FOM (top) and POD-Galerkin-DEIM (bottom) displacements computed at different times for $\boldsymbol{\mu} = [11.4625 \text{ Pa}]$.

2.3.2 Benchmark problems with a prolate spheroid geometry

In this section we investigate the performances of a POD-Galerkin-DEIM reduced order model to address benchmark problems in cardiac mechanics. In both cases, the reference geometry $\Omega_0 \subset \mathbb{R}^3$ is that of a truncated ellipsoid and the material law adopted is the nearly-incompressible Guccione relation (1.3), although different parameter values are taken into account, according to [Lan+15]. For what concerns the boundary condition of problem (1.2), we apply a linear external pressure

$$\mathbf{g}(t; \boldsymbol{\mu}) = \widetilde{p}\, t/T$$

at the endocardium (i.e. Γ_0^N), with $\widetilde{p} > 0$, simulating the presence of blood inside the cardiac chamber, and consider homogeneous Neumann and Dirichlet conditions at the epicardium (i.e. Γ_0^R with $\alpha = \beta = 0$) and on the base (i.e Γ_0^D), respectively, the latter representing the artificial boundary resulting from truncation of the heart below the valves in a short axis plane.

The FOM is built on a hexahedral mesh with 4804 elements and 6455 vertices, depicted in Figure 2.15, resulting in a high-fidelity dimension $N_h = 19365$, since $\mathbb{Q}_1$-FE (that is, linear FE on a hexahedral mesh) are used.

Passive inflation of a left ventricle

In this first case we do not account for anisotropy and only consider the passive contribution of the Piola tensor, so that the obtained deformation pattern simulates passive ventricular diastole.

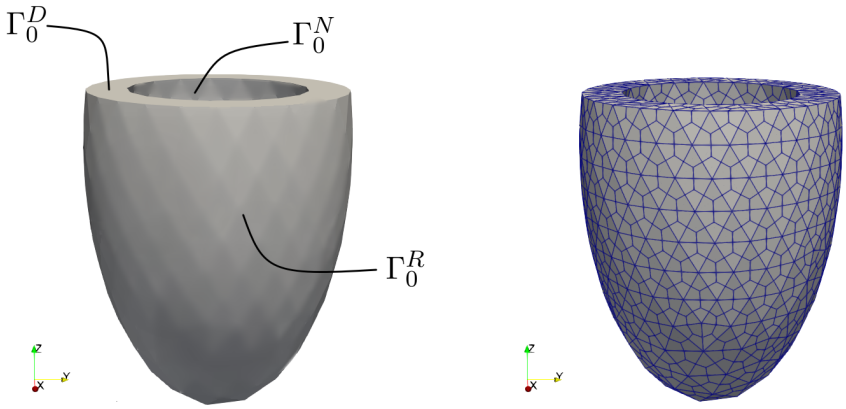


Figure 2.15: Idealized truncated ellipsoid geometry (left) and computational grid (right).

As unknown parameters, we consider

- the material stiffness in fiber and cross-fiber directions $b_f, b_n \in [0.8, 1.2]$, and
- the multiplicative factor $C \in [8 \cdot 10^3, 12 \cdot 10^3]$ Pa.

All other parameters are fixed to their reference values, namely $b_s = b_{fs} = b_{fn} = b_{sn} = 1$, $K = 50 \cdot 10^3$ Pa and $\widetilde{p} = 10 \cdot 10^3$. For the time setting, we choose $t \in (0, 0.25)$ s and a uniform time step $\Delta t = 5 \cdot 10^{-3}$ s, resulting in a total number of 50 time instances. In Figure 2.16 we report the high-fidelity displacement at different time instants. In order to compute the FOM solution, almost 340 s are required in average on a PC desktop computer with 3.70GHz Intel Core i5-9600K CPU and 16GB RAM.

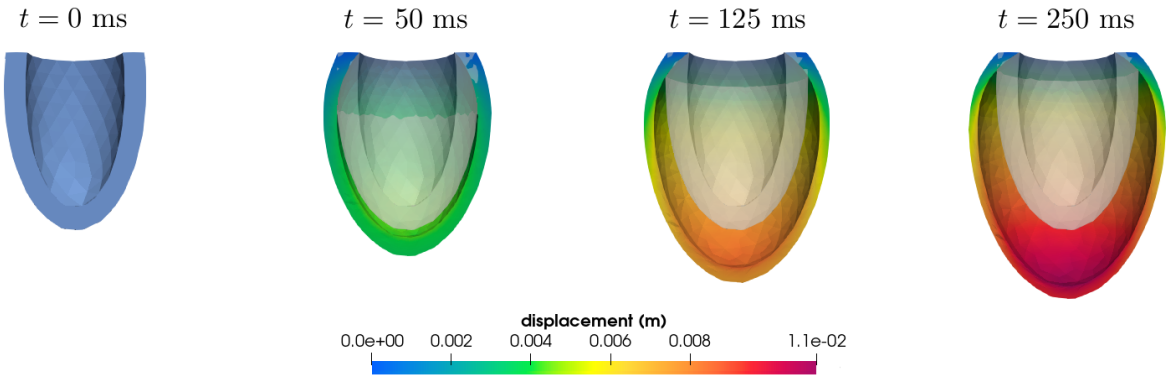


Figure 2.16: Passive inflation of a left ventricle. FOM solution computed at different times for $\boldsymbol{\mu} = [1.007, 0.853, 8610 \text{ Pa}]$.

For $n_s = 50$ points sampled in the parameter space $\mathcal{P}$ through LHS, we collect the solution snapshots in order to compute the RB basis $\mathbf{V} \in \mathbb{R}^{N_h \times N}$. The singular values arising from the SVD of the snapshots matrix $\mathbf{S}_u$ are reported in Figure 2.17, where a rapid decay of the reported quantity is observed. We expect that a small number of basis functions is sufficient for the ROM to guarantee a good approximation of the high-fidelity solution manifold.

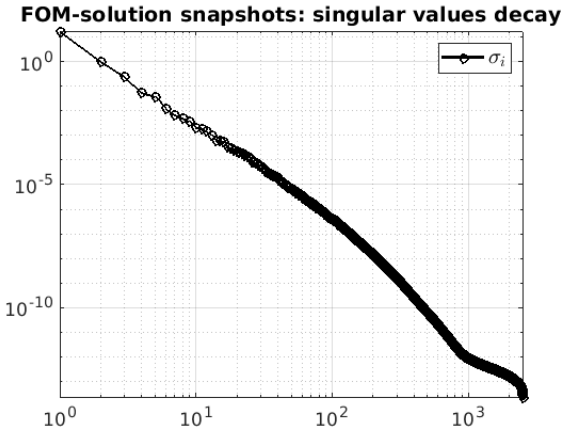


Figure 2.17: Passive inflation of a left ventricle. Decay of the singular values of the FOM solution snapshots matrix.

As done in previous test case to analyze the performances of the ROM without hyper-reduction, we consider the following POD tolerances:

$$\varepsilon_{POD} \in \{10^{-3}, 5 \cdot 10^{-4}, 10^{-4}, 5 \cdot 10^{-5}, 10^{-5}, 5 \cdot 10^{-6}, 10^{-6}\},$$

corresponding to a RB dimension of $N = 5$, 7, 12, 16, 26, 32 and 55, respectively. Figure 2.18 shows the error and the CPU time ratio over ε_{POD}, where the average is computed over a testing set of 20 parameters, different from those used for the generation of FOM snapshots. The plot confirms that few basis functions are required to accurately approximate the high-fidelity displacements. In particular, we observe that the relative error ϵ_{rel} associated with the ROM approximation decreases of almost three orders of magnitude when going from $N = 5$ to $N = 55$. However, since at each Newton step we need to assemble high-dimensional arrays before projecting them onto the reduced space, the CPU time required by the ROM for all dimensions N is almost comparable to the one of the FOM.

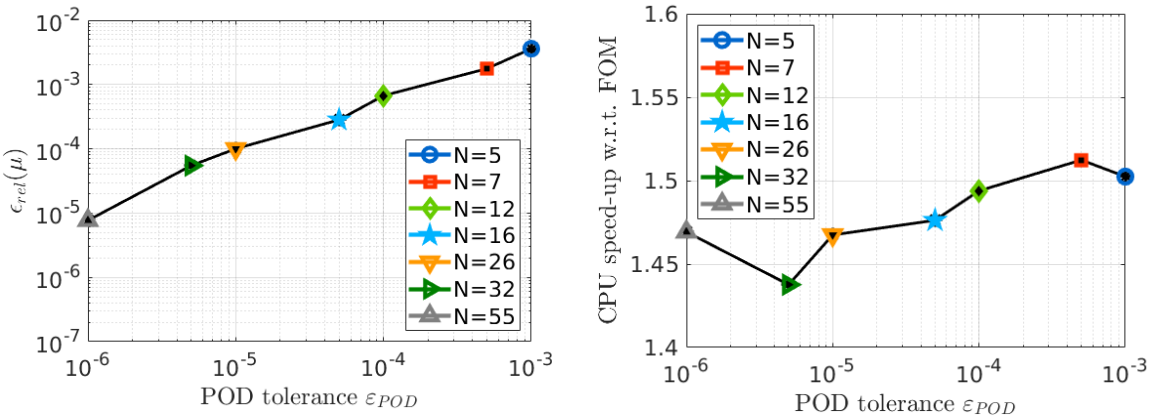


Figure 2.18: Passive inflation of a left ventricle. Average over 20 testing parameters of relative error ϵ_{rel} (left) and speed-up (right) of POD-Galerkin ROMs.

Further, we investigate the impact of hyper-reduction onto the ROM solution reconstruction error. In order to build the POD-Galerkin-DEIM hyper-reduced model, we consider $N = 16$ basis functions for state reduction, obtained for $\varepsilon_{POD} = 10^{-5}$. For the construction of the residual basis $\mathbf{\Phi}_{\mathcal{R}} \in \mathbb{R}^{N_h \times m}$, we rely on a snapshots set computed for $n'_s = 50$ points in $\mathcal{P}$ and apply POD on $\mathbf{S}_R$ using

$$\varepsilon_{DEIM} \in \{5 \cdot 10^{-4}, 10^{-4}, 5 \cdot 10^{-5}, 10^{-5}, 5 \cdot 10^{-6}, 10^{-6}\},$$

corresponding to $m = 84$, 127, 148, 207, 239 and 368, and compare these options. As shown in Figure 2.19 and Table 2.3, no loss of accuracy is experienced when decreasing the number of DEIM interpolation points m. Higher tolerances ε_{DEIM} on the DEIM error do not ensure the convergence of the reduced Newton system for all testing parameters, such that no higher speed-ups can be achieved by further reducing the size of the residual basis. Furthermore, we point out that $m > N$, meaning that the residual vector shows higher variability with respect to the displacement. To conclude, the POD-Galerkin-DEIM hyper-reduced order model with $m = 84$ is able to achieve a speed-up of $\times 9.3$ compared to the FOM, still achieving an approximation error equals to the projection error, that is $\epsilon_{rel} \approx 10^{-3}$.

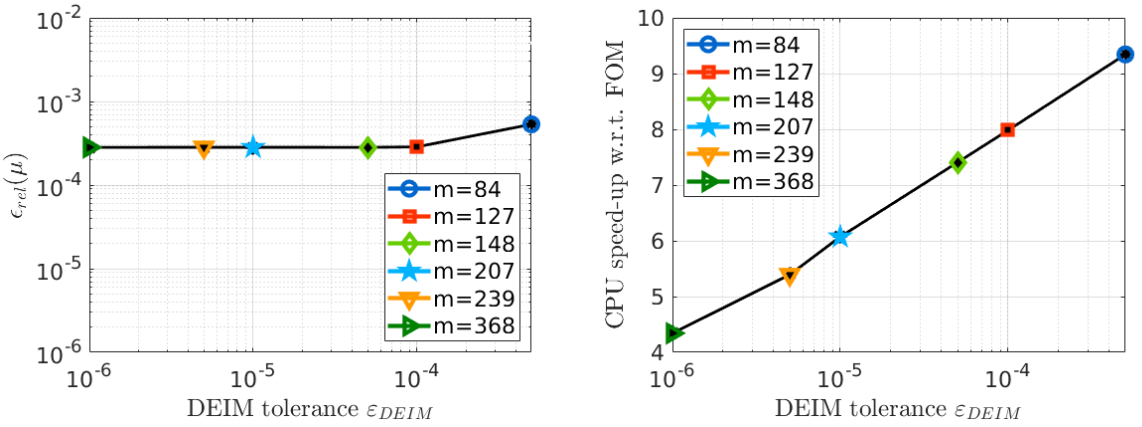


Figure 2.19: Passive inflation of a left ventricle. Average over 20 testing parameters of relative error ϵ_{rel} (left) and average speed-up (right) of POD-Galerkin-DEIM with $N = 16$ and different values of m.

	$m = 84$	$m = 127$	$m = 368$
Computational speed-up	$\times 9.3$	$\times 8.0$	$\times 4.3$
Avg. CPU time	36 s	42 s	77 s
Time-avg. $L^2(\Omega_0)$-absolute error	$5.3 \cdot 10^{-5}$	$2.9 \cdot 10^{-5}$	$2.9 \cdot 10^{-5}$
Time-avg. $L^2(\Omega_0)$-relative error	$5.3 \cdot 10^{-4}$	$2.9 \cdot 10^{-4}$	$2.8 \cdot 10^{-4}$

Table 2.3: Passive inflation of a left ventricle. Computational data related to POD-Galerkin-DEIM with $N = 16$ and different values of m.

Passive inflation and active contraction of a left ventricle

The second benchmark takes into account both a varying fiber distribution and contractile forces, dealing with the inflation and the active contraction of an idealized left ventricle with transversely isotropic material properties. Despite being an idealized test case, the displacement field reproduces the typical twisting motion of the systole in the left ventricle, caused by the distribution of the muscular fibers.

In order to compute the fiber orientation in cardiac geometries, suitable rule-based methods have been developed [Bay+12; WK14; Dos+19; Pie+21], which usually depend on a set of parameter angles. In this particular case we consider the method proposed in [Ros+14], where $\boldsymbol{\alpha}^{epi}$ and $\boldsymbol{\alpha}^{endo}$ represent possible values of the rotation angle of the fibers on the epicardium and endocardium, respectively. Even if electrophysiology in not taken into account, we need to surrogate the presence of the active generation forces that drive the contraction mechanics. With this aim, active contraction is modeled through the active stress approach, considering anisotropic active tension applied in the fiber direction only, see Section 1.2.2. In particular, the parameterized active tension in the fiber direction $T_a = T_a(t;\boldsymbol{\mu})$ is modeled as a linear function of the form

$$T_a(t;\boldsymbol{\mu}) = \widetilde{T}_a \; t/T.$$

Since we want to assess the performance of a POD-Galerkin-DEIM reduced order model to reduce the myocardium contraction, we consider as unknown parameters those related to the active components of the strain energy function:

- the maximum value of the active tension $\widetilde{T}_a \in [49.5 \cdot 10^3, 70.5 \cdot 10^3]$ Pa, and
- the fiber angles $\boldsymbol{\alpha}^{epi} \in [-105.5, -74.5]^\circ$ and $\boldsymbol{\alpha}^{endo} \in [74.5, 105.5]^\circ$.

All other parameters are fixed to their reference values, namely $b_f = 8$, $b_s = b_n = b_{sn} = 2$, $b_{fs} = b_{fn} = 4$, $C = 2 \cdot 10^3$ Pa, $K = 50 \cdot 10^3$ Pa and $\widetilde{p} = 15 \cdot 10^3$. Regarding the time discretization, we choose $t \in (0, 0.25)$ s and a uniform time step $\Delta t = 5 \cdot 10^{-3}$ s.

Given a training set of $n_s = 50$ points obtained by sampling the parameter space $\mathcal{P}$, we perform a convergence analysis of the ROM without hyper-reduction by constructing the reduced basis $\mathbf{V} \in \mathbb{R}^{N_h \times N}$ for different N and computing the associated approximation errors.

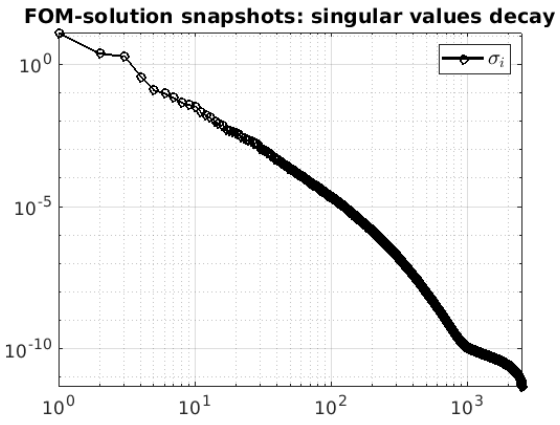


Figure 2.20: Passive inflation and active contraction of a left ventricle. Decay of the singular values of the FOM solution snapshots matrix.

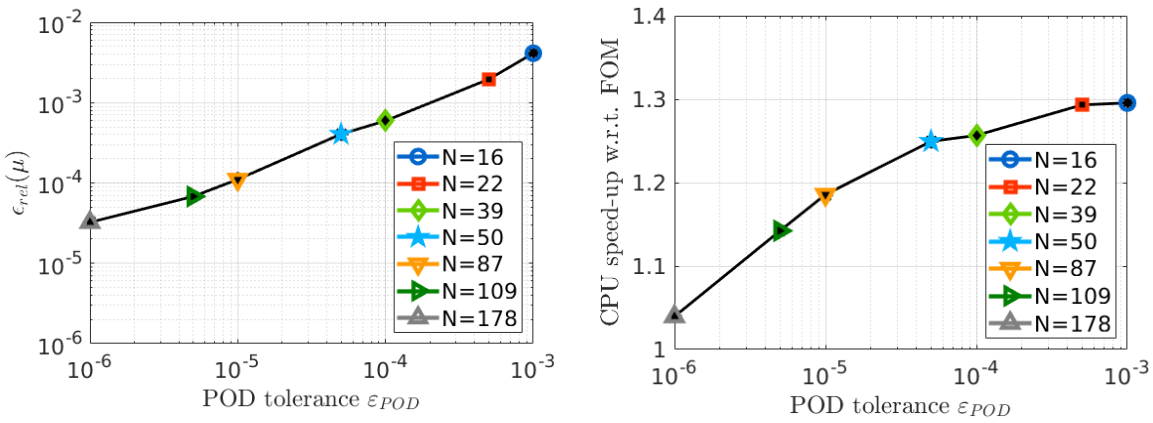


Figure 2.21: Passive inflation and active contraction of a left ventricle. Average over 20 testing parameters of relative error ϵ_{rel} (left) and speed-up (right) of POD-Galerkin ROMs.

From Figure 2.20, we observe a slower decay of the singular values of $\mathbf{S}_u$ with respect to the previous test case of passive inflation. In fact, using POD tolerances

$$\varepsilon_{POD} \in \{10^{-3}, 5 \cdot 10^{-4}, 10^{-4}, 5 \cdot 10^{-5}, 10^{-5}, 5 \cdot 10^{-6}, 10^{-6}\},$$

we obtain the RB dimensions $N = 16$, 22, 39, 50, 87, 109 and 178, respectively, much larger than the ones obtained for similar tolerances on the previous test case. This behavior is somehow expected, as the underlying system dynamics, simulating ventricular contraction and associated torsion, is more involved than the idealized diastole, in which tissue anisotropy is neglected. As shown by the results obtained so far, the construction of a ROM is highly problem dependent, since the parameters considered, e.g., in the constitutive relation, strongly influence the form of the solution manifold, thus the dimension of the reduced basis necessary to obtain comparable accuracy between the different ROMs.

The error and the CPU speed-ups averaged over a testing set of 20 parameters are both shown in Figure 2.21, as functions of the POD tolerance ε_{POD}. As already discussed, the speed-ups achieved by the ROMs are negligible, since at each Newton iteration without hyper-reduction the ROMs still depend on the high-fidelity dimension N_h. For what concerns the approximation error, we observe a reduction of almost two orders of magnitude when going from $N = 16$ to $N = 178$.

Given $\mathbf{V} \in \mathbb{R}^{N_h \times N}$ with $N = 16$, we construct POD-Galerkin-DEIM approximations by considering $n'_s = 200$ parameter samples. We point out that smaller values of n'_s ($n'_s \geq 50$) can be considered, leading to almost the same results in the online phase. Figure 2.22 shows the decay of the singular values of $\mathbf{S}_R$, that is, the snapshots matrix of the residual vectors $\mathbf{R}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}_{\ell'}), t^n; \boldsymbol{\mu}_{\ell'})$. We observe that the reported curve decreases very slowly, so that we expect that a large number of basis functions is required to correctly approximate the nonlinear operators.

In fact, by computing $\boldsymbol{\Phi}_{\mathcal{R}} \in \mathbb{R}^{N_h \times m}$ using the following POD tolerances:

$$\varepsilon_{DEIM} \in \{5 \cdot 10^{-4}, 10^{-4}, 5 \cdot 10^{-5}, 10^{-5}, 5 \cdot 10^{-6}, 10^{-6}\},$$

we obtain $m = 303$, 456, 543, 776, 902 and 1233, respectively. Higher values of ε_{DEIM} (related to hopefully smaller dimensions m) were not sufficient to guarantee the convergence of the

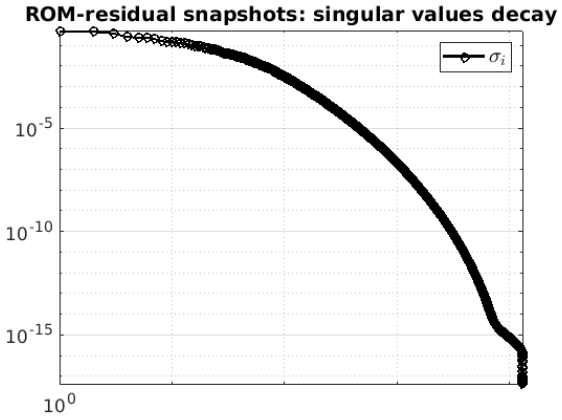


Figure 2.22: Passive inflation and active contraction of a left ventricle. Decay of the singular values of the ROM (N=16) residual snapshots matrix.

reduced Newton problem for all the parameter combinations considered. The average relative errors over a set of 20 parameters and the computational speed-ups are reported in Figure 2.23. In particular, we observe that the relative error is between $4 \cdot 10^{-3}$ and $8 \cdot 10^{-3}$, as we could expect from the projection error reported in Figure 2.21.

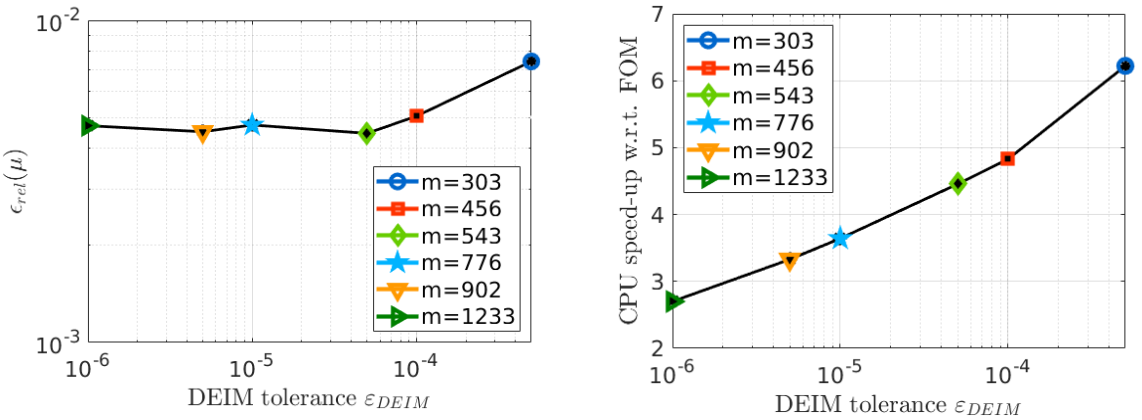


Figure 2.23: Passive inflation and active contraction of a left ventricle. Average over 20 testing parameters of relative error ϵ_{rel} (left) and average speed-up (right) of POD-Galerkin-DEIM with $N = 16$.

Similar results are obtained when using a finer mesh with 13025 vertices, corresponding to $N_h = 39075$ dofs for the FOM. In this case a reduced basis of dimension $N = 16$ is computed using $\varepsilon_{POD} = 10^{-3}$, while the POD tolerance ε_{DEIM} for the approximation of the nonlinear arrays has to be chosen no larger than 10^{-4} to ensure Newton convergence, obtaining $m = 385$ DEIM terms and a corresponding approximation error $\epsilon_{rel} \approx 5 \cdot 10^{-3}$. Also in this case, the highest speed-up achieved by the ROM is greater than that obtained for the coarser mesh (i.e. $\times 8$ against $\times 6$), suggesting that the POD-Galerkin-DEIM convenience grows as the dimension of the underlying FOM increases.

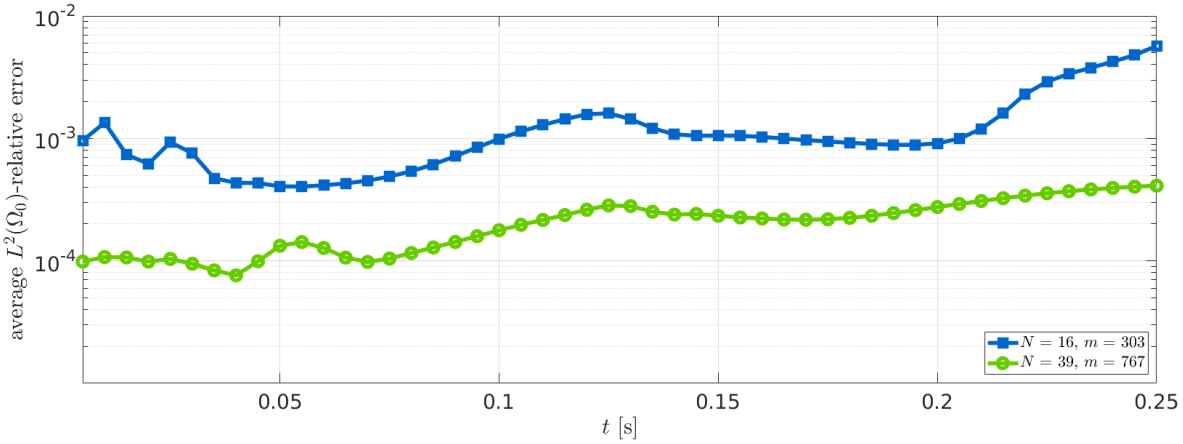


Figure 2.24: Passive inflation and active contraction of a left ventricle. Evolution in time of the average $L^2(\Omega_0)$-relative error for $(N, m) = (16, 303)$ (blue) and $(N, m) = (39, 767)$ (greed).

Indeed, when a larger reduced basis is considered, such as in the case of $N = 39$ basis functions obtained for $\varepsilon_{POD} = 10^{-4}$ when $N_h = 19365$, a remarkable improvement in accuracy is achieved, as shown in Figure 2.24 and Table 2.4. However, a higher number of DEIM magic points is required for the solution of the reduced nonlinear system through Newton method, ultimately doubling the online CPU time required to solve the hyper-ROM for each parameter instance.

ε_{POD}	10^{-3} $(N = 16)$	10^{-4} $(N = 39)$
ε_{DEIM}	$5 \cdot 10^{-4}$ $(m = 303)$	10^{-5} $(m = 767)$
Computational speed-up	$\times 6.2$	$\times 2.9$
Avg. CPU time	58 s	124 s
Time-avg. $L^2(\Omega_0)$-absolute error	$1 \cdot 10^{-3}$	$2 \cdot 10^{-4}$
Time-avg. $L^2(\Omega_0)$-relative error	$7 \cdot 10^{-3}$	$1 \cdot 10^{-3}$

Table 2.4: Passive inflation and active contraction of a left ventricle. Computational data related to POD-Galerkin-DEIM with $(N, m) = (16, 303)$ and $(N, m) = (39, 767)$.

As a matter of fact, the residual basis dimension m is highly influenced by the size N of the reduced subspace for the solution, so that a larger basis $\mathbf{V}$ requires a greater number of interpolation points to correctly approximate the reduced nonlinear operators, thus reducing the overall speed-up of the ROM. Therefore, choosing $N = 16$ represents a good trade off between accuracy and efficiency. The high-fidelity solution and the POD-Galerkin-DEIM reduced displacement computed using $m = 303$ at different time instants are shown in Figures 2.25 and 2.26 for two parameter values, together with their point-wise difference. About 6 minutes are required in average to solve the high-fidelity problem for each instance of the parameter, while the solution of the hyper-ROM requires less than a minute, thus achieving a speed-up equal to $\times 6.2$.

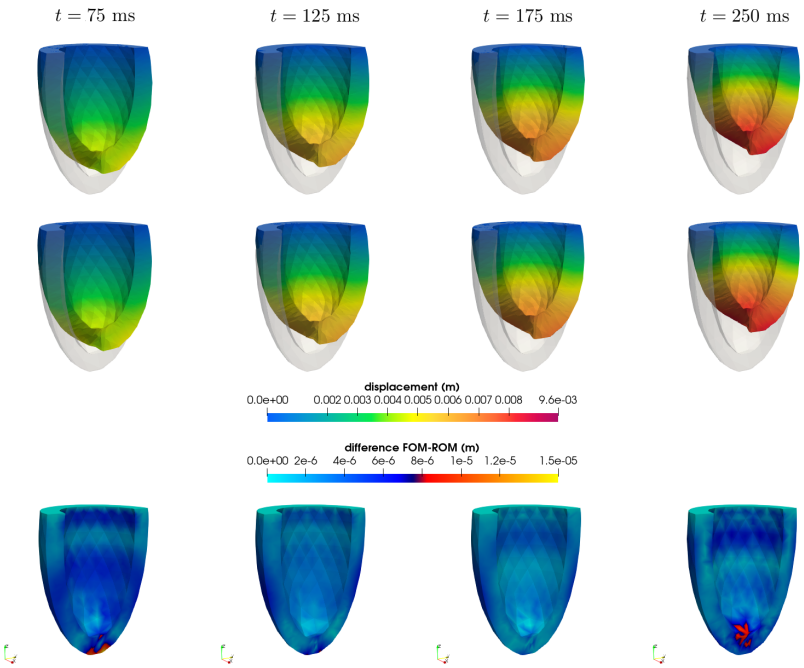


Figure 2.25: Passive inflation and active contraction of a left ventricle. FOM (top) and POD-Galerkin-DEIM (middle) displacements at different times for $\boldsymbol{\mu} = [61942.5 \text{ Pa}, -77.5225°, 87.9075°]$ and difference between them (bottom), for $N = 16$.

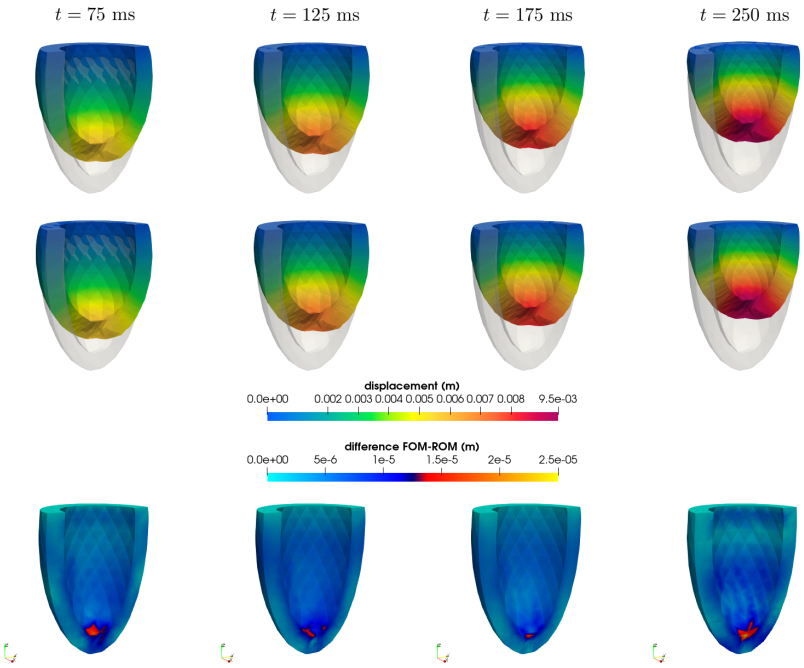


Figure 2.26: Passive inflation and active contraction of a left ventricle. FOM (top) and POD-Galerkin-DEIM (middle) displacements at different times for $\boldsymbol{\mu} = [59737.5 \text{ Pa}, -102.3225°, 91.1625°]$ and difference between them (bottom), for $N = 16$.

The case of high dimensional parameter spaces At last, we consider a higher number of training parameters, that are,

- the material stiffness in different directions $b_f \in [6.6, 9.4]$, $b_s, b_n, b_{sn} \in [1.65, 2.35]$, $b_{fs}, b_{fn} \in [3.3, 4.7]$,
- the bulk modulus $K \in [4 \cdot 10^4, 6 \cdot 10^4]$ Pa,
- the multiplicative constant $C \in [1.5 \cdot 10^3, 2.5 \cdot 10^3]$ Pa,
- the maximum active tension $\widetilde{T}_a \in [49.5 \cdot 10^3, 70.5 \cdot 10^3]$ Pa,
- the steepness of the pressure ramp $\widetilde{p} \in [14 \cdot 10^3, 16 \cdot 10^3]$ Pa, and
- the fiber angles $\boldsymbol{\alpha}^{epi} \in [-105.5, -74.5]^\circ$ and $\boldsymbol{\alpha}^{endo} \in [74.5, 105.5]^\circ$,

such that,

$$\boldsymbol{\mu} = \left[b_f, b_s, b_n, b_{fs}, b_{fn}, b_{sn}, K, C, \widetilde{T}_a, \widetilde{p}, \boldsymbol{\alpha}^{epi}, \boldsymbol{\alpha}^{endo}\right] \in \mathcal{P} \subset \mathbb{R}^{12}.$$

For the construction of the FOM solution and the ROM residual snapshots matrices, we consider $n_s = 50$ and $n_s' = 75$ training parameters, respectively. Further details are summarized in Table 2.5. We observe that, despite having increased the number of unknown parameters from 3 to 12, the dimension N of the RB functions is almost unaffected by the choice of a higher dimensional parameter space, as the mechanical behavior is mostly influenced by the active parameters $\widetilde{T}_a$, $\boldsymbol{\alpha}^{epi}$ and $\boldsymbol{\alpha}^{endo}$ previously considered.

ε_{POD}	1e-3 ($N = 19$)
ε_{DEIM}	1e-4 ($m = 574$)
Computational speed-up	$\times 4.1$
Avg. CPU time	87 s
Time-avg. $L^2(\Omega_0)$-absolute error	$3 \cdot 10^{-3}$
Time-avg. $L^2(\Omega_0)$-relative error	$2 \cdot 10^{-3}$

Table 2.5: Passive inflation and active contraction of a left ventricle. Computational data related to POD-Galerkin-DEIM with $N = 19$ and $m = 574$, when size($\boldsymbol{\mu}$) = 12.

2.3.3 Idealized full cycle of the left ventricle with a patient specific geometry

Finally, we want to analyze the performances of the POD-Galerkin-DEIM method when the whole cardiac cycle is taken into account. For the sake of simplicity, we employ suitable analytical time-dependent functions to surrogate the influence of the active stress model and blood circulation. In particular, the active tension $\mathbf{T}_a(\mathbf{X}, t; \boldsymbol{\mu}) = T_a(t; \boldsymbol{\mu})(\mathbf{F}\mathbf{f}_0 \otimes \mathbf{f}_0)$ and the pressure $\mathbf{g}(t; \boldsymbol{\mu})$ are computed as follows:

1. for a fixed set of physiological parameters, we solve the cardiac electromechanics (EM) problem in $(0, T)$ coupled with a closed-loop 0D circulation model, as done in [Reg+20a],

where the active forces are modeled by means of an artificial neural network built from biophysically detailed high-fidelity models [RDQ20b];

2. we compute the space-average of the active tension coming from the EM simulation, that is
$$\tau_a^{EM}(t) = \operatorname*{avg}_{\mathbf{X}\in\Omega_0} T_a^{EM}(\mathbf{X}, t),$$
where $\mathbf{T}_a^{EM}(\mathbf{X}, t) = T_a^{EM}(\mathbf{X}, t)(\mathbf{F}\mathbf{f}_0 \otimes \mathbf{f}_0)$ is the active stress acting in the fiber direction, as well as the EM blood pressure $\mathbf{g}^{EM}(t)$;

3. by performing a cubic spline interpolation of $\tau_a^{EM}(t)$ and $\mathbf{g}^{EM}(t)$, we obtain the corresponding analytical surrogate functions $\tau_a^M(t)$ and $\mathbf{g}^M(t)$, reported in Figure 2.27 for $T = 0.8$ s;

4. finally, to take into account a parameter-dependence, we define
$$T_a(t; \boldsymbol{\mu}) = \frac{\widetilde{T}_a}{\max\limits_{t\in(0,T)} \tau_a^M(t)} \tau_a^M(t), \qquad \mathbf{g}(t; \boldsymbol{\mu}) = \frac{\widetilde{p}}{\max\limits_{t\in(0,T)} \mathbf{g}^M(t)} \mathbf{g}^M(t),$$
in order to model different maximum active tensions and loading conditions.

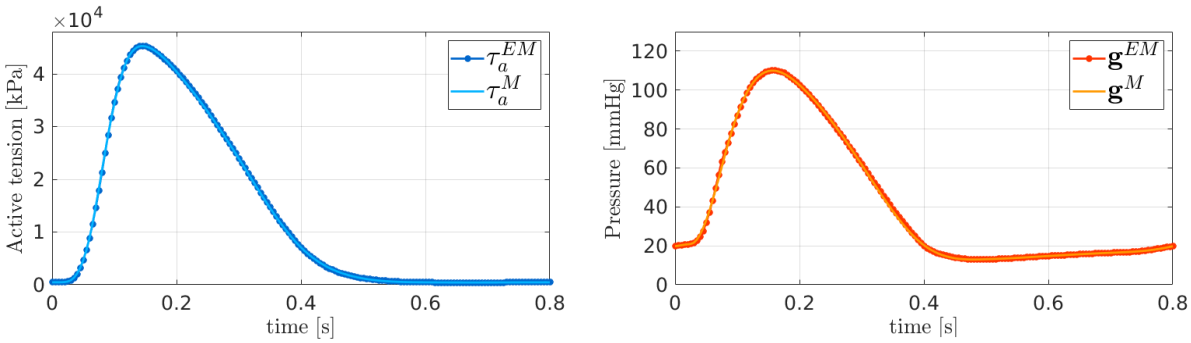


Figure 2.27: Idealized full cycle of the left ventricle. Space-averaged active tension (left) and blood pressure (right) computed during a EM simulation and the corresponding fitted curves.

The reference geometry $\Omega_0 \subset \mathbb{R}^3$ is a patient-specific left ventricle, pre-processed from the Zygote Solid 3D heart model [Inc14] reconstructed from an high resolution computed tomography scan, whose associated hexahedral mesh with 6167 vertices is reported in Figure 2.28. The FOM is built using $\mathbb{Q}_1$-FE, such that the high-fidelity dimension is $N_h = 18501$. Differently from all other test cases considered so far, we assume the following boundary conditions:

- Robin boundary conditions at the epicardium Γ_0^{epi} (or Γ_0^R)
$$\mathbf{P}(\mathbf{u})\mathbf{N} = -(\mathbf{N}\otimes\mathbf{N})(K_\perp \mathbf{u} + C_\perp \partial_t \mathbf{u}) - (\mathbf{I} - \mathbf{N}\otimes\mathbf{N})(K_\parallel \mathbf{u} + C_\parallel \partial_t \mathbf{u}),$$
see Table .6 for the values of $K_\perp, K_\parallel, C_\perp$ and $C_\parallel$;

- Neumann boundary conditions at the endocardium Γ_0^{endo} (or Γ_0^N)
$$\mathbf{P}(\mathbf{u})\mathbf{N} = -\mathbf{g}(t; \boldsymbol{\mu}) J \mathbf{F}^{-T} \mathbf{N};$$

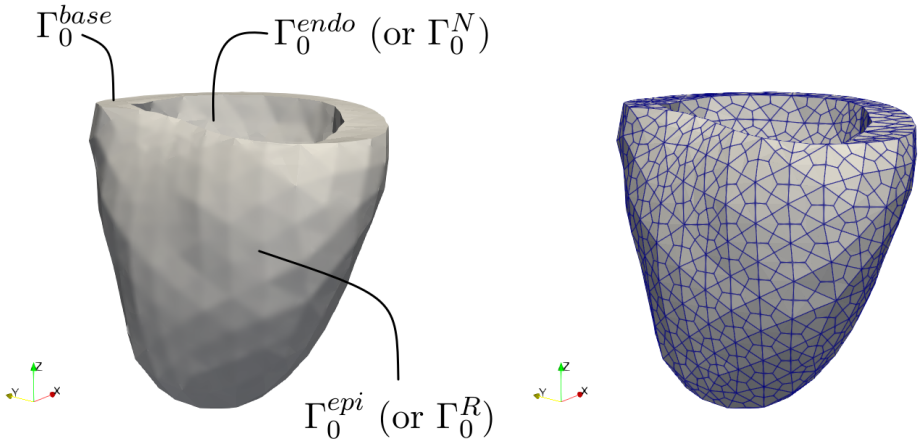


Figure 2.28: Patient-specific unloaded left ventricle geometry (left) and computational grid (right).

- energy-consistent boundary conditions [RDQ20a] at the base Γ_0^{base}

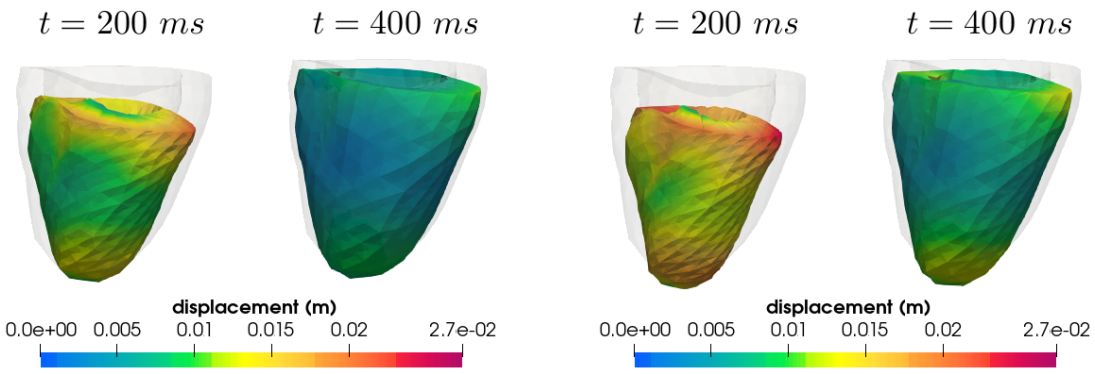
$$\mathbf{P}(\mathbf{u})\mathbf{N} = -\frac{\|J\mathbf{F}^{-T}\mathbf{N}\|}{\int_{\Gamma_0^{base}} \|J\mathbf{F}^{-T}\mathbf{N}\| d\Gamma} \int_{\Gamma_0^{endo}} \mathbf{P}(\mathbf{u})\mathbf{N} d\Gamma.$$

We point out that this choice of boundary conditions do not affect the construction of the ROM, since no assumptions about the form of the full model are made in the reduction strategy. As for the previous test cases, we consider the nearly-incompressible Guccione law and adopt the active stress approach. Regarding the fiber distribution, we employ the rule-based method proposed in [Bay+12], depending on parameter angles $\boldsymbol{\alpha}^{epi}$, $\boldsymbol{\alpha}^{endo}$, $\boldsymbol{\beta}^{epi}$ and $\boldsymbol{\beta}^{endo}$. For time discretization, we consider a uniform time step $\Delta t = 5 \cdot 10^{-3}$ s and set $T = 0.8$ s, corresponding to the duration of a single heartbeat, resulting in a total number of 160 time iterations. We choose, as unknown parameters, those most affecting cardiac deformation during systole and diastole, that are,

- the multiplicative factor of the constitutive law $C \in [0.44 \cdot 10^3, 1.32 \cdot 10^3]$ Pa,
- the active tension parameter $\widetilde{T}_a \in [49.5 \cdot 10^3, 70.5 \cdot 10^3]$ Pa, and
- the fiber angle at the epicardium $\boldsymbol{\alpha}^{epi} \in [-75, -45]°$.

All other parameters are fixed to their reference values, namely $b_f = 8$, $b_s = 6$, $b_n = b_{fn} = b_{sn} = 3$, $b_{fn} = 12$, $K = 50 \cdot 10^3$ Pa, $\widetilde{p} = 15 \cdot 10^3$ Pa, $\boldsymbol{\alpha}^{epi} = 60°$, $\boldsymbol{\beta}^{epi} = 20°$ and $\boldsymbol{\beta}^{endo} = -20°$. In Figure 2.29 we report the FOM solution at different time instants and for two values of the parameter vector $\boldsymbol{\mu} = [C, \widetilde{T}_a, \boldsymbol{\alpha}^{epi}]$. Computing the high-fidelity solution for each instance of the parameter required almost 15 minutes on a PC desktop computer with 3.70GHz Intel Core i5-9600K CPU and 16GB RAM.

Remark 2.5
Due to the fact that we are neglecting the coupling between mechanics and circulation, non-physiological pressure-volume loops are obtained, especially in the first time steps of the heartbeat, corresponding to ventricular systole, which lacks of the isometric contraction phase.



(a) $\boldsymbol{\mu} = [1.078, 50.025, -66.75]$ **(b)** $\boldsymbol{\mu} = [0.682, 69.975, -54.75]$

Figure 2.29: Idealized full cycle of the left ventricle. FOM solution computed at different times.

In addition to this, the end-diastolic configuration of the left ventricle should be recovered before starting the mechanics simulation. However, we point out that our aim is to test and analyze the reduction methodology on parameterized time-dependent nonlinear PDEs. Moreover, the simulated cardiac cycle can be regarded as a sufficiently accurate reproduction of both systole and diastole deformations for the purpose at hand.

For the construction of the reduced basis $\mathbf{V}$, we collect $n_S = 20$ FOM solutions and perform POD with $\varepsilon_{POD} = 10^{-3}$, obtaining a reduced subspace of dimension $N = 28$. To build the residual basis, we perform $n'_s = 50$ ROM simulations and collect the residual snapshots. Since we are using $N_t = 160$ and $n'_s = 50$, and at least two Newton iterations are performed at each time step, we end up with a residual snapshots matrix $\mathbf{S}_R$ of more than 16000 columns. For this reason, we rely on randomized-SVD to speed-up the computation of $\boldsymbol{\Phi}_{\mathcal{R}}$, by choosing a priori the number of DEIM basis, rather the the POD tolerance ε_{DEIM} (see Algorithm 3). Table 2.6 summarizes the average errors computed over a testing set of 20 parameters and the

	$m = 850$	$m = 1000$	$m = 1200$
Computational speed-up	$\times 3.0$	$\times 2.8$	$\times 2.7$
Avg. CPU time	285 s	303 s	311 s
Time-avg. $L^2(\Omega_0)$-absolute error	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$
Time-avg. $L^2(\Omega_0)$-relative error	$8 \cdot 10^{-3}$	$8 \cdot 10^{-3}$	$8 \cdot 10^{-3}$

Table 2.6: Idealized full cycle of the left ventricle. Computational data related to POD-Galerkin-DEIM with $N = 28$ and different values of m.

CPU times obtained using $m = 850$, 1000 and 1200, while volumes and pressure-volume loops for $m = 850$ are reported in Figure 2.30. Here, POD-Galerkin-DEIM outputs are compared to those of the FOM for two different values of the parameter. We point out that no convergence of the reduced Newton system has been obtained for all testing parameters when using smaller residual basis. In Figures 2.31 and 2.32 we report the POD-Galerkin-DEIM solutions at four

time instants, for two different values of the parameter vector $\boldsymbol{\mu} = [C, \widetilde{T}_a, \boldsymbol{\alpha}^{epi}]$, respectively. Moreover, the corresponding pointwise difference between the high-fidelity solution and its reduced order approximation is also reported.

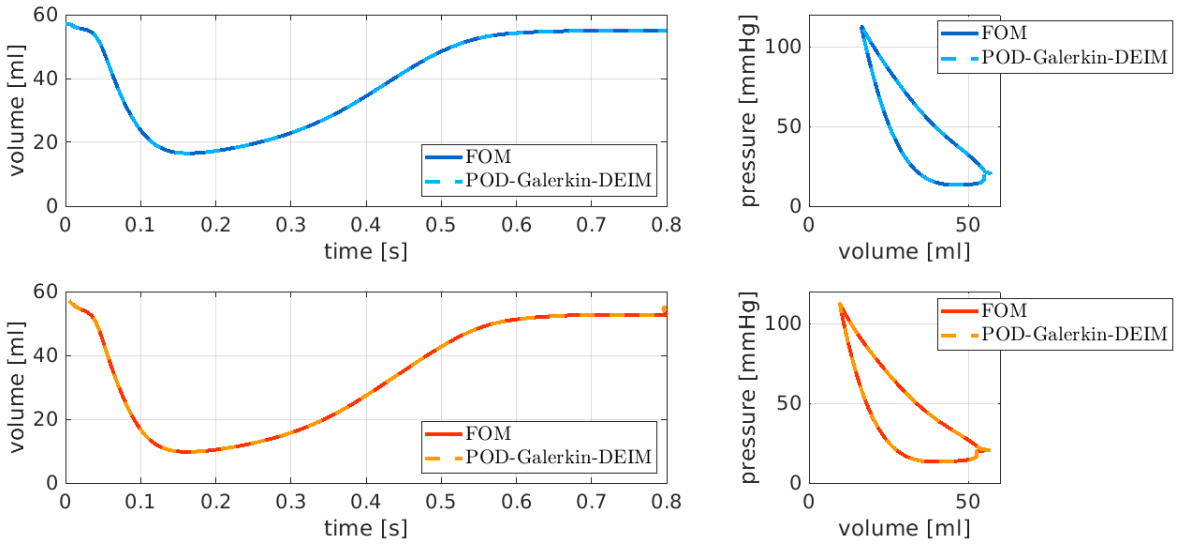


Figure 2.30: Idealized full cycle of the left ventricle. Ventricular cavity volume (left) and pressure-volume loop (right) for parameter values $\boldsymbol{\mu} = [1.078, 50.025, -66.75]$ (top) and $\boldsymbol{\mu} = [0.682, 69.975, -54.75]$ (bottom).

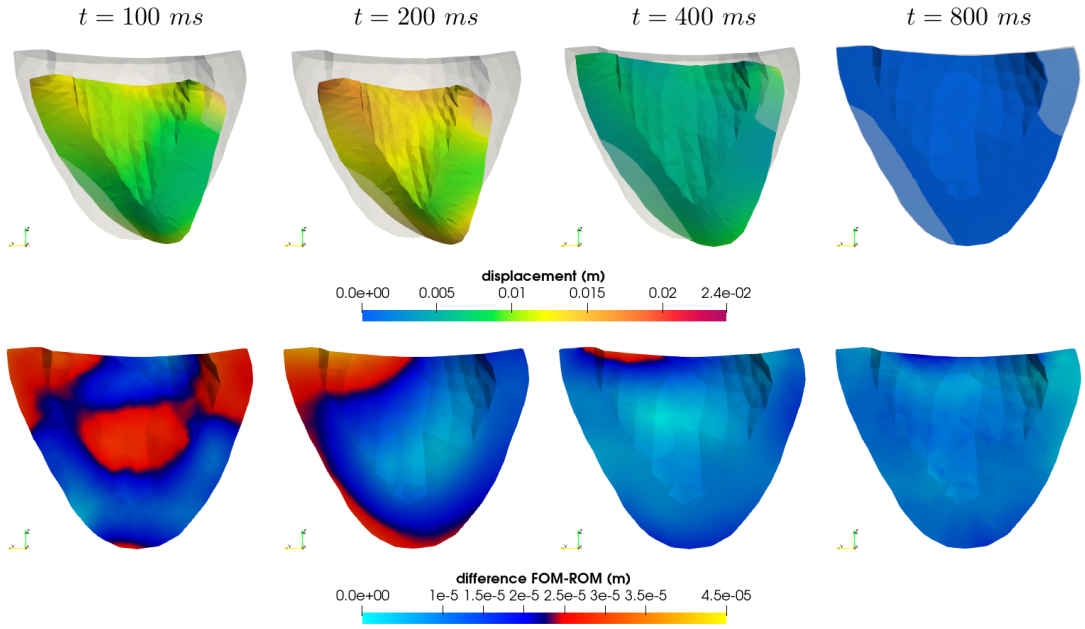


Figure 2.31: Idealized full cycle of the left ventricle. Patient specific geometry of a left ventricle. POD-Galerkin-DEIM solution (top) and error (bottom) computed at different times for $\boldsymbol{\mu} = [1.078, 50.025, -66.75]$.

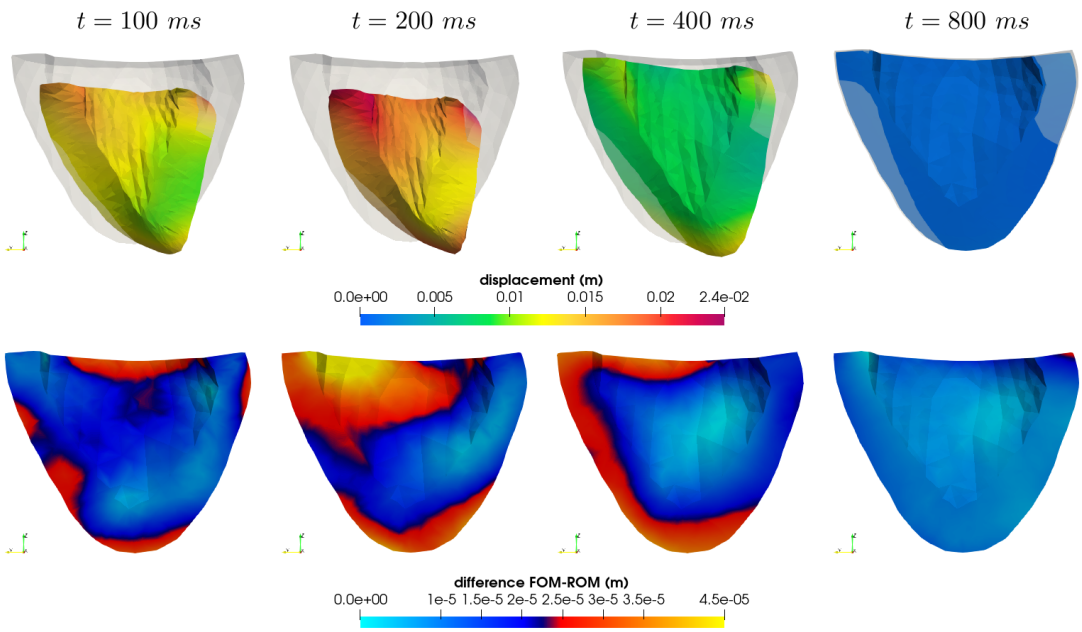


Figure 2.32: Idealized full cycle of the left ventricle. Patient specific geometry of a left ventricle. POD-Galerkin-DEIM solution (top) and error (bottom) computed at different times for $\boldsymbol{\mu} = [0.682, 69.975, -54.75]$.

2.4 The hyper-reduction bottleneck

The POD-Galerkin method, equipped with DEIM hyper-reduction, proved able to accurately approximate the high-fidelity solution of parameterized nonlinear time-dependent problems, such as those reproducing different phases of the cardiac cycle on both idealized and realistic geometries. So far, few works have addressed the reduction to solid mechanics problems using the RB method and DEIM as hyper-reduction technique, e.g., [RR16; GTS17]. Indeed, we noticed that a small number of RB functions is required to build a good approximation of the reduced subspace for all considered test cases, and this is a distinguishing feature of several parameterized problems in elastodynamics. However, the computational speed-up achieved by the ROM decreases as the complexity of the underlying problem (with respect to either the parameter space, the geometrical mesh or the IBVP itself) increases. The main difficulty arises from the approximation of ROM residual vectors, which feature a greater variability with respect to the solution, especially in the reduction of problems arising in cardiac mechanics, where an exponentially nonlinear constitutive law is considered. As a matter of fact, even if the nonlinear quantities are assembled onto a reduced mesh, a large number of interpolation points is needed to guarantee the convergence of the reduced Newton system for complex applications, thus compromising the efficiency of the POD-Galerkin-DEIM method, as most of the online CPU time is required for assembling the reduced residuals.

The data reported in Table 2.7 for the performances of the POD-Galerkin-DEIM technique applied to the active contraction benchmark problem, with $N = 16$ and $\boldsymbol{\mu} = [\widetilde{T}_a, \boldsymbol{\alpha}^{epi}, \boldsymbol{\alpha}^{endo}]$, show that the computational bottleneck of the method is indeed the construction of the reduced system at each Newton iteration, which requires more than 90% of the online CPU time, while

the actual reduced system solution is performed very efficiently (0.01%). In particular, almost

POD tolerance ε_{DEIM}	$5 \cdot 10^{-4}$	$5 \cdot 10^{-5}$	$5 \cdot 10^{-6}$
DEIM interpolation dofs m	303	543	902
Reduced mesh elements (total: 4804)	914	1345	1855
Online CPU time	58 s	81 s	110 s
◦ system construction [∗]	89%	93%	94%
◦ system solution	0.01%	0.01%	0.01%
[∗] System construction for each Newton iteration	0.4 s	0.6 s	0.9 s
◦ residual assembling	94%	94%	94%
◦ Jacobian computing through AD	0.24%	0.24%	0.26%
Computational speed-up	×6.2	×4.5	×3.3
Time-averaged $L^2(\Omega_0)$-absolute error	$1 \cdot 10^{-3}$	$7 \cdot 10^{-4}$	$6 \cdot 10^{-4}$
Time-averaged $L^2(\Omega_0)$-relative error	$8 \cdot 10^{-3}$	$5 \cdot 10^{-3}$	$5 \cdot 10^{-3}$

Table 2.7: Passive inflation and active contraction of a left ventricle. Computational data related to the DEIM approach with $N = 16$ and different values of m.

94% of the computational time required for system construction at each Newton iteration is used for assembling the residual $\mathbf{R}(\mathbf{V}\mathbf{u}_N^{n+1,(k)}, t^{n+1}; \boldsymbol{\mu})$ on the reduced mesh. Instead, the computation of the associated Jacobian through AD is extremely fast and requires less than 0.3% of the CPU time, such that, we cannot rely on quasi-Newton methods to improve the online performances. The remaining CPU time is mostly used for matrix multiplication. Similar results were found as well for the simpler problem of the deforming beam (see Table 2.1).

This study suggests the idea of using surrogate models to perform operator approximation, overcoming the need to assemble the nonlinear terms onto a computational mesh; this will be the object of Chapter 4, where a new technique relying on deep learning algorithms to approximate operators is proposed, which also represents one of the main contributions of this Thesis.

Chapter 3

Uncertainty quantification in structural mechanics

To provide reliable approximations of physical phenomena described by means of PDEs, understanding how uncertainty in the model inputs affects the considered outputs, as well as estimating the input parameters from available data, are key tasks. As shown in Chapter 2, model input parameters may have a large impact on the solution to the problem under investigation, so that it is crucial to characterize their influence on output quantities of interest that are useful, e.g., in the clinical practice [RCSR19; Mar+20]. The knowledge of the parameter values of cardiac models is often severely limited due to both the difficulties in performing experiments [Kov+21] and the uncertainty associated with their measurements. Moreover, some parameters are related to the mathematical model and do not have a direct physical meaning. whilst others are hampered by intrinsic variability among patients [Mir+16].

Identification and propagation of uncertainties and parameters calibration are related to the field of uncertainty quantification (UQ), which can be divided into two groups of problems: forward UQ and inverse UQ. The aim of the former is to determine statistical information about the uncertainty in outputs of interest given statistical information about the uncertainty in the inputs; the goal of the latter is to infer about unknown input parameters based on observed outputs.

The solution to both types of problems requires the repeated evaluation of the input-output map, so that relying on high-fidelity FOMs for the approximation of the underlying PDE solution may become computationally unfeasible. In order to reduce the computational costs, surrogate models are commonly adopted [HCM17; Hau+18; RCSR19; Cam+20]. In this chapter we show how to take advantage of ROMs, such as those based on POD-Galerkin-DEIM, for the performance of sensitivity analysis studies and Bayesian calibration in the field of structural mechanics. Indeed, the efficient solution to these UQ problems represent a further step in the construction of patient-specific models that can be possibly used to improve diagnosis, provide real-time decision support and predict prognosis.

In Section 3.1 we introduce the basic concepts of variance-based global sensitivity analysis (SA) and provide few numerical examples about the mutual utility of SA and reduced order models, while Section 3.2 is devoted to the solution of parameter estimation problems in a Bayesian framework. The theoretical concepts introduced in this Chapter are mainly based on [Sal+08; Smi13; Sul15], to which we refer for further details.

3.1 Variance-based global sensitivity analysis

The goal of SA is to quantify the effects of parameter variation on one or many quantities of interest, in order to provide a criterion to rank the inputs with respect to their influence on the outputs. Sensitivities are usually discovered by running the underlying model against different configurations and analyzing the statistical properties of the associated input-output samples. Among different tasks of SA, we mention:

- factor prioritization (FP), which aims at identifying which factors, once fixed at their true values, lead to the greatest reduction in the variance of the outputs, so that these identified factors should be properly measured, and
- factor fixing (FF), whose goals is to identify which of the inputs can be fixed anywhere in their range of variation without noticeably affecting the output of interest. These factors can thus be arbitrarily fixed in order to simplify the model.

For these reasons, SA studies are usually run in tandem with UQ and may also be useful in the development of reduced order models; indeed, avoiding to include inessential (or less relevant) parameters among the model inputs can enhance the complexity and the training stage of the ROMs.

The approaches to perform SA can be broadly classified into two groups according to the nature of the information provided, that are local and global methods. The former are based on the calculation of partial derivatives of the outputs with respect to the inputs, so that the information provided are related to a neighborhood of the points at which derivatives are evaluated. The latter, instead, are based on sampling techniques on the entire parameter domain. Since a handful of random data points is more informative and robust than estimating derivatives at a single data point, in this Thesis we focus on the latter class of methods, particularly on variance-based global sensitivity analysis.

3.1.1 Sobol' indices

Let $(\boldsymbol{\Xi}, \mathcal{F}, \mathbb{P})$ be a probability space and consider the random variables

$$\begin{aligned} M_i \colon \boldsymbol{\Xi} &\to \mathbb{R} \\ \xi &\mapsto \mu_i = M_i(\xi), \end{aligned}$$

for $i = 1, \dots, P$, so that $\mathbf{M} = (M_1, \dots, M_P)$ denotes the random parameter vector and $\boldsymbol{\mu} = (\mu_1, \dots, \mu_P)$ its realization for a given $\xi \in \boldsymbol{\Xi}$. For the sake of simplicity, we assume all the input parameters to be independent and uniformly distributed, and normalized in the range $[0, 1]$. From a black-box perspective, a generic model can be seen as a function of the random vector $\mathbf{M}$, that is

$$Y = y(\mathbf{M}),$$

where the model realization $y(\boldsymbol{\mu})$ is, e.g., the output of interest associated with the PDE solution computed for $\boldsymbol{\mu}$ as input parameter. Here Y is a univariate model output, but multiple outputs can be analyzed by conducting independent sensitivity studies. Note that y is defined over the P-dimensional unit hypercube; thus, if y is integrable, it can be decomposed into summands of

increasing dimensions as

$$y = y_0 + \sum_i y_i + \sum_{i<j} y_{ij} + \ldots + y_{1\ldots P},$$

where each term is a function of the factors in its index. Under the assumption that each term in the above expansion has zero mean, we can univocally calculate these terms as [Sob90]:

$$\begin{aligned} y_0 &= \mathbb{E}[Y], \\ y_i &= \mathbb{E}[Y|M_i] - y_0, \\ y_{ij} &= \mathbb{E}[Y|M_i, M_j] - y_i - y_j - y_0, \end{aligned}$$

and so on. Further assuming that y is square integrable, leads to the so-called ANOVA high-dimensional model representation (or ANOVA-HDMR decomposition)

$$\mathrm{Var}(Y) = \sum_i V_i + \sum_{i<j} V_{ij} + \ldots + V_{1\ldots P}, \tag{3.1}$$

where

$$V_i = \mathrm{Var}(y_i) = \mathrm{Var}(\mathbb{E}[Y|M_i]),$$

for $i = 1, \ldots, P$, are the first-order effects,

$$V_{ij} = \mathrm{Var}(y_{ij}) = \mathrm{Var}(\mathbb{E}[Y|M_i, M_j]) - V_i - V_j,$$

for $i, j = 1, \ldots, P$ $(i < j)$, are the second-order effects, and so on. This decomposition shows how the (total) variance of the model output can be decomposed into terms related to each input and to the interactions between them. Variance-based measures represent a powerful tool to quantify the relative importance of individual factors or groups. Our aim is to determine what happens to the uncertainty of Y when a factor M_i is fixed to a particular value μ_i.

First order effects

A primary variance-based measure of sensitivity, referred to as *first-order sensitivity index* or *first Sobol' index*, is given by

$$S_i = \frac{V_i}{\mathrm{Var}(Y)} = \frac{\mathrm{Var}(\mathbb{E}[Y|M_i])}{\mathrm{Var}(Y)}, \quad i = 1, \ldots, P.$$

This index represents the main contribution of the input factor M_i to the variance of the output, as it measures the effect of varying M_i alone averaged over variations in all input parameters. First Sobol' indices are useful to perform factor prioritization (FP), as the higher is S_i, the greater is the influence of M_i on the output Y.

To better see this, let μ_i be a particular value for M_i and consider the conditional variance

$$\mathrm{Var}(Y|M_i = \mu_i),$$

that is the variance of Y taken over all factors but M_i. In order to make this measure independent of μ_i, we take its expectation. According to the law of total variance, we can write

$$\mathbb{E}[\mathrm{Var}(Y|M_i)] = \mathrm{Var}(Y) - \mathrm{Var}(\mathbb{E}[Y|M_i]), \tag{3.2}$$

which, divided by $\mathrm{Var}(Y)$, gives

$$\frac{\mathbb{E}[\mathrm{Var}(Y|M_i)]}{\mathrm{Var}(Y)} = 1 - S_i.$$

Since the left-hand side of (3.2) corresponds to the model variance once a source of variation has been fixed, we understand that the more influential is M_i, the smaller is $\mathbb{E}[\mathrm{Var}(Y|M_i)]$, hence the greater must be S_i.

Total effects

If a model is non-additive, so that it is not possible to separate the effects of its inputs on the output, it holds

$$\sum_i S_i < 1$$

and we should look for higher order interactions. In particular, by dividing both sides of (3.1) by $\mathrm{Var}(Y)$, we obtain

$$\sum_i S_i + \sum_{i<j} S_{ij} + \ldots + S_{1\ldots P} = 1.$$

However, performing such full sensitivity analysis requires the evaluation of $2^P - 1$ indices, which can be computationally demanding, especially when the input-output map $\boldsymbol{\mu} \to y(\boldsymbol{\mu})$ is highly nonlinear and it is not possible to compute the expectation values and the variances in close form, so that Monte Carlo (MC) methods should be applied.

To avoid the computation of all the interaction terms, we can rely on the total effect indices S_{T_i}, for $i = 1, \ldots, P$, which account for the contribution to the variance of Y due to factor M_i, including all its interactions with the other factors. Let

$$\mathbf{M}_{\sim i} = (M_1, \ldots, M_{i-1}, M_{i+1}, \ldots, M_P)$$

be the random vector of all input factors but M_i, for $i = 1, \ldots, P$. From the law of total variance, we can write

$$\mathrm{Var}(Y) = \mathbb{E}[\mathrm{Var}(Y|\mathbf{M}_{\sim i})] + \mathrm{Var}(\mathbb{E}[Y|\mathbf{M}_{\sim i}]), \tag{3.3}$$

where $\mathbb{E}[\mathrm{Var}(Y|\mathbf{M}_{\sim i})]$ corresponds to the (average) variance of Y left if $\mathbf{M}_{\sim i}$ could be fixed to their true values. Dividing (3.3) by $\mathrm{Var}(Y)$, we obtain the so-called *total-effect index* or *total-order index* for M_i, defined as

$$S_{T_i} = \frac{\mathbb{E}[\mathrm{Var}(Y|\mathbf{M}_{\sim i})]}{\mathrm{Var}(Y)} = \frac{\mathrm{Var}(Y) - \mathrm{Var}(\mathbb{E}[Y|\mathbf{M}_{\sim i}])}{\mathrm{Var}(Y)} = 1 - \frac{\mathrm{Var}(\mathbb{E}[Y|\mathbf{M}_{\sim i}])}{\mathrm{Var}(Y)},$$

for $i = 1, \ldots, P$. The smaller is S_{T_i}, the less influential is M_i. A necessary and sufficient condition for a factor M_i to be non-influential is that its total Solbol' index is zero, i.e. $S_{T_i} = 0$, so that M_i can be arbitrarily fixed within its range of uncertainty without appreciably affecting the output of interest Y. Total indices are thus well suited for factor fixing (FF) to reduce the number of input parameter for the model. Note that, unlike S_i, it holds

$$\sum_i S_{T_i} \geq 1,$$

since $S_{T_{i_1}}$ and $S_{T_{i_2}}$, for $i_1 < i_2$, take both into account the interactions between M_{i_1} and M_{i_2}.

Numerical computation of the sensitivity indices

For most functions y, the sensitivity indices described in the sections above cannot be calculated analytically and must be estimated by suitable techniques, such as MC or quasi-MC methods. In this section we describe the procedure developed in [Sal02] (based on the approaches originally proposed in [Sob90; HS96]), which avoids the cumbersome computation of multidimensional integrals by brute-force. Let $N_S \in \mathbb{N}$ be a prescribed integer, known as *base sample*. Sobol' method (or Saltelli method) consists in the following steps:

1. generate a $N_S \times 2P$ matrix of random parameter realizations obtained, e.g., from a Sobol' quasi-random sequence [Sob67];

2. define two matrices of data $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{N_S \times P}$ each containing half of the sample, i.e.

$$\mathbf{A} = \begin{bmatrix} \mu_1^{(a,1)} & \dots & \mu_P^{(a,1)} \\ \vdots & & \vdots \\ \mu_1^{(a,N_S)} & \dots & \mu_P^{(a,N_S)} \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} \mu_1^{(b,1)} & \dots & \mu_P^{(b,1)} \\ \vdots & & \vdots \\ \mu_1^{(b,N_S)} & \dots & \mu_P^{(b,N_S)} \end{bmatrix},$$

   where μ_i is the realization of the i-th random variable M_i.

3. construct P matrices $\mathbf{C}_i$, $i = 1, \dots, P$, formed by all columns of $\mathbf{B}$ except the i-th column taken from $\mathbf{A}$, i.e.

$$\mathbf{C} = \begin{bmatrix} \mu_1^{(b,1)} & \dots & \mu_{i-1}^{(b,1)} & \mu_i^{(a,1)} & \mu_{i+1}^{(b,1)} & \dots & \mu_P^{(b,1)} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ \mu_1^{(b,N_S)} & \dots & \mu_{i-1}^{(b,N_S)} & \mu_i^{(a,N_S)} & \mu_{i+1}^{(b,N_S)} & \dots & \mu_P^{(b,N_S)} \end{bmatrix}$$

4. compute the output of the model for all the parameter vectors given by the rows of $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}_i$, for $i = 1, \dots, P$, thus obtaining the N_S-dimensional vectors

$$\mathbf{y}_A = y(\mathbf{A}), \quad \mathbf{y}_B = y(\mathbf{B}), \quad \mathbf{y}_{C_i} = y(\mathbf{C}_i).$$

5. for $i = 1, \dots, P$, estimate the i-th first order sensitivity index as follows,

$$S_i = \frac{\mathbf{y}_A \cdot \mathbf{y}_{C_i} - \bar{\mathbf{y}}_A^2}{\mathbf{y}_A \cdot \mathbf{y}_A - \bar{\mathbf{y}}_A^2},$$

   where $\bar{\mathbf{y}}_A = \dfrac{1}{N_S} \sum\limits_{i=1}^{N_S} \mathbf{y}_A^{(i)}$ is the mean.

6. finally, for $i = 1, \dots, P$, estimate the i-th total order index as follows,

$$S_{T_i} = 1 - \frac{\mathbf{y}_B \cdot \mathbf{y}_{C_i} - \bar{\mathbf{y}}_A^2}{\mathbf{y}_A \cdot \mathbf{y}_A - \bar{\mathbf{y}}_A^2}.$$

An explanation on why these formulas work can be found, e.g, in [Sal+08]. For P input factors, computing a full set of S_i's and S_{T_i}'s indices requires $N_S(P+2)$ runs of the model to evaluate $\mathbf{y}_A$, $\mathbf{y}_B$ and $\mathbf{y}_{C_i}$, for $i = 1, \dots, P$. On the other hand, N_S^2 runs would have been needed in the brute-force approach. However, since the accuracy of the estimates depends on N_S, which can vary from few hundreds to few thousands, the main drawback of this method is still its computational cost, especially when the model is expensive to evaluate. When this is the case, reduced order models can be employed to greatly alleviate the computational expense.

3.1.2 Application of reduced order models to sensitivity analysis

So far we have considered $Y = y(\mathbf{M})$ to be a continuous function of the parameter random vector $\mathbf{M}$. However, when y is related to a PDE problem, one usually needs to rely on numerical approximation techniques in order to find a discrete solution $\mathbf{u}_h = [\mathbf{u}_h^1, \dots, \mathbf{u}_h^{N_t}]$, so that, given a realization $\boldsymbol{\mu}$ of the parameter random vector $\mathbf{M}$, the corresponding model output is given by

$$\mathbf{y}_h(\boldsymbol{\mu}) = Q(\mathbf{u}_h(\boldsymbol{\mu})),$$

where $Q\colon \mathbb{R}^{N_h \times N_t} \to \mathbb{R}^{N_q}$ is the vector-valued function of the quantities of interest. We recall that, when $N_q > 1$, different sensitivity analysis can be carried out, one for each output.

In this setting the ROMs developed in Chapter 2, providing the reduced input-output map

$$\boldsymbol{\mu} \mapsto \mathbf{u}_N(\boldsymbol{\mu}) \mapsto Q(\mathbf{u}_N(\boldsymbol{\mu})) = \mathbf{y}_N(\boldsymbol{\mu})$$

at a low cost computational, can be used to speed-up the evaluation of Sobol' indices [PM]. In particular, we can replace $\mathbf{y}_h$ with $\mathbf{y}_N$ in step 4 of the Saltelli method, thus efficiently obtaining the reduced output vectors

$$\mathbf{y}_A = \mathbf{y}_N(\mathbf{A}), \quad \mathbf{y}_B = \mathbf{y}_N(\mathbf{B}), \quad \mathbf{y}_{C_i} = \mathbf{y}_N(\mathbf{C}_i),$$

which are used in the computation of Sobol' indices S_i and S_{T_i}, for $i = 1, \dots, P$. Note that, in this case, the reliability of the approximation depends both on the base sample N_S and on the accuracy of the ROMs. However, thanks to the lower computational cost of the reduced models, a higher number of samples can be used, thus balancing the discrepancy between the high-fidelity and the reduced input-output maps.

In the following subsections we focus on applications, dealing with a simple rectangular beam geometry that undergoes a large deformation due to a pressure load applied on the surface, and a benchmark problem for cardiac mechanics reproducing ventricular systole. In this case we consider an idealized left ventricle, given by an truncated ellipsoid.

Numerical example: deformation of a neo-Hookean clamped rectangular beam

Let $\bar{\Omega}_0 = [0, 10^{-2}] \times [0, 10^{-3}] \times [0, 10^{-3}]$ m be the reference geometry and consider the structural problem of a clamped beam with an external load applied at the bottom face (see Section 2.3.1). For the sake of convenience, we recall that pressure boundary conditions with external load

$$\mathbf{g}(t; \boldsymbol{\mu}) = \widetilde{p}\, t/T, \quad t \in (0, T),$$

are applied to $\Gamma_0^N = \partial\Omega_0^{\{z=0\}}$ (see Figure 3.1), Dirichlet boundary conditions are imposed on the left face $x = 0$ m, and homogeneous Neumann conditions are considered for all other faces.

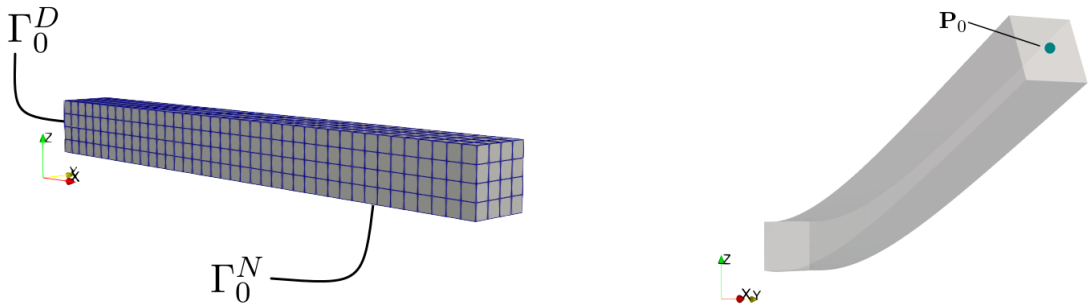


Figure 3.1: Deformation of a neo-Hookean clamped rectangular beam. Computational mesh (left) and point $\mathbf{P}_0$ (right) for the evaluation of the quantities of interest.

As final simulation time we choose $T = 0.25$ s. The FOM is built on a hexahedral mesh with 1025 vertices using $\mathbb{Q}_1$-FE for the approximation in space and a uniform partition of the time interval with $\Delta t = 5 \cdot 10^{-3}$ s, thus resulting in a high-fidelity dimension $N_h = 3075$ and in a total number of 50 time steps.

For the first test case, we consider a nearly-incompressible neo-Hookean material characterized by the following strain density energy function

$$\mathcal{W}(\mathbf{F}) = \frac{G}{2}(J^{-\frac{2}{3}} \det(\mathbf{C}) - 3) + \frac{K}{4}((J-1)^2 + \ln^2(J)),$$

where $G > 0$ is the shear modulus and K is the bulk modulus. The parameter vector $\boldsymbol{\mu}$ is thus given by:

- the shear modulus $G \in [0.5 \cdot 10^4, 1.5 \cdot 10^4]$ Pa,
- the bulk modulus $K \in [2.5 \cdot 10^4, 7.5 \cdot 10^4]$ Pa, and
- the external load parameter $\widetilde{p} \in [2, 6]$ Pa,

so that $P = 3$. The choice of this numerical test featuring a small number P of input parameters and a low complexity of the FOM is motivated by the fact that we want to compare the results of the SA performed using high-fidelity and reduced order models. Since $N_S(P+2)$ model runs are required to compute the Sobol' indices of the inputs, where the sample base N_S should be chosen sufficiently big to ensure accuracy, using expensive full order models may become computationally unfeasible.

As quantity of interest we choose the displacement in the vertical direction of a point $\mathbf{P}_0$ at different time instants. To be more specific, we choose $\mathbf{P}_0$ such that its material coordinate is $\mathbf{X}_{P_0} = (10^{-2}, 0.5 \cdot 10^{-3}, 0.5 \cdot 10^{-3})$, thus corresponding to the central point of the right face $x = 10^{-2}$ m of Γ_0^N (i.e. opposite to Γ_0^D), as shown in Figure 3.1, and set

$$\mathbf{y}_{h,j}(\boldsymbol{\mu}) = Q_j(\mathbf{u}_h(\boldsymbol{\mu})) = \mathbf{u}_h(\mathbf{X}_{P_0}, t^{10j}; \boldsymbol{\mu}) \cdot \mathbf{e}_z,$$

for $j = 1, \ldots, \lfloor N_t/10 \rfloor$, so that $N_q = \lfloor N_t/10 \rfloor = 5$. Note that $\mathbf{y}_{h,j}$ denotes the j-th component of high-fidelity output vector $\mathbf{y}_h(\boldsymbol{\mu}) = Q(\mathbf{u}_h(\boldsymbol{\mu})) \in \mathbb{R}^{N_q}$.

Since we are considering a multivariate model for the output (i.e. $N_q > 1$), multiple SAs are run in parallel, such that we end up with N_q vectors of first-order and total sensitivity indices

$$\mathbf{S}^j = \left[S_1^j, \dots, S_P^j\right] \quad \text{and} \quad \mathbf{S}_T^j = \left[S_{T_1}^j, \dots, S_{T_P}^j\right],$$

respectively, for $j = 1, \dots, N_q$.

In order to compare the accuracy of the Sobol' indices obtained by replacing the high-fidelity input-output map

$$\boldsymbol{\mu} \mapsto \mathbf{u}_h(\boldsymbol{\mu}) \mapsto \mathbf{y}_h(\boldsymbol{\mu})$$

with the reduced input-output map

$$\boldsymbol{\mu} \mapsto \mathbf{u}_N(\boldsymbol{\mu}) \mapsto \mathbf{y}_N(\boldsymbol{\mu}),$$

we construct a POD-Galerkin-DEIM ROM with $N = 4$ POD basis functions and $m = 22$ DEIM basis functions, as done in Section 2.3.1, and run Saltelli method with the base sample equal to $N_s = 1000$, so that 5000 model evaluations are required. We recall that in this case the FOM takes 26 s on a PC desktop computer with 3.70GHz Intel Core i5-9600K CPU and 16GB RAM for each new instance of the parameter, while the ROM requires on average 2.4 s, so that around 36 h and 3 h are required in total when using the FOM and the ROM, respectively. All numerical simulations are performed using the Dakota toolkit [Ada+20] and its Python direct interface. Table 3.1 reports Sobol' indices computed using the FOM and the ROM for each quantity of interest, showing that the FOM can be replaced by a more efficient reduced model without affecting the results of the sensitivity analysis.

Index →		S_i^1	S_i^2	S_i^3	S_i^4	S_i^5	S_T^1	S_T^2	S_T^3	S_T^4	S_T^5
G	FOM	0.185	0.490	0.433	0.407	0.451	0.203	0.529	0.467	0.438	0.483
	ROM	0.188	0.493	0.435	0.406	0.454	0.205	0.534	0.468	0.437	0.486
K	FOM	0.009	0.017	0.017	0.016	0.016	0.004	0.008	0.008	0.007	0.007
	ROM	0.010	0.017	0.017	0.016	0.017	0.004	0.008	0.008	0.007	0.008
$\widetilde{p}$	FOM	0.787	0.417	0.478	0.523	0.472	0.787	0.450	0.501	0.543	0.492
	ROM	0.785	0.410	0.477	0.525	0.466	0.785	0.445	0.498	0.545	0.487

Table 3.1: Deformation of a neo-Hookean clamped rectangular beam. Sensitivity indices computed using the FOM and POD-Galerkin-DEIM.

By computing the average over the sensitivity indices, we obtain

$$\bar{\mathbf{S}} = [0.4, 0.02, 0.5], \quad \bar{\mathbf{S}}_T = [0.4, 0.01, 0.6]$$

and we can conclude that G and $\widetilde{p}$ are the most influential factors, so that they are more likely to be identified when solving a parameter estimation problem (see in Section 3.2), whereas the bulk modulus K has a small impact on the quantities of interest considered and can be fixed at an arbitrary value within its domain of variability to reduce the model complexity.

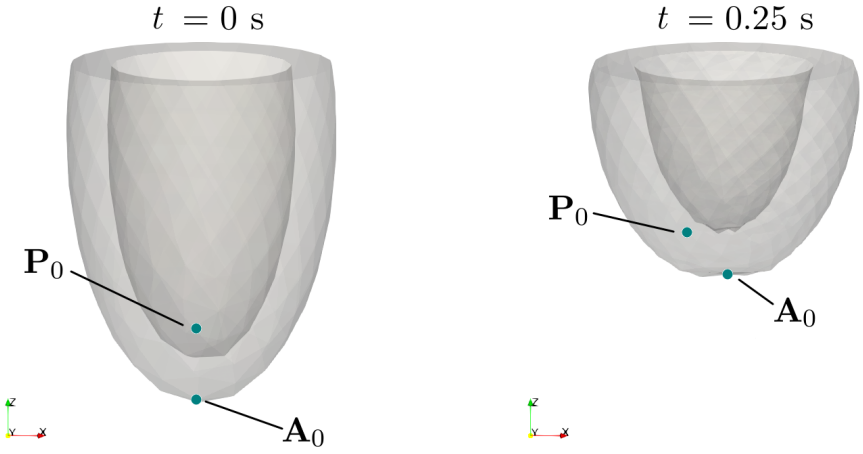


Figure 3.2: Passive inflation and active contraction of a left ventricle. Apex $\mathbf{A}_0$ and point $\mathbf{P}_0$.

Numerical example: passive inflation and active contraction of a left ventricle

We now consider the benchmark problem of the passive inflation and the active contraction of an idealized left ventricle. In this case a linear external pressure

$$\mathbf{g}(t;\boldsymbol{\mu}) = \widetilde{p}\, t/T, \quad t \in (0,T)$$

is applied at the endocardium to simulate the presence of blood inside the ventricle, the base is kept fixed and homogeneous Neumann boundary conditions are imposed at the epicardium. We consider a Guccione material and take into account contractile forces by introducing the parameterized active tension

$$\mathbf{T}_a(t;\boldsymbol{\mu}) = \widetilde{T}_a\, t/T\, (\mathbf{F}\mathbf{f}_0 \otimes \mathbf{f}_0), \quad t \in (0,T),$$

in the constitutive law, where $T = 0.25$ s is the final simulation time and $\mathbf{f}_0$ is the reference unit vector in the fiber direction. The FOM is built on a hexahedral mesh with 6455 vertices using $\mathbb{Q}_1$-FE, such that $N_h = 19365$, and employing a uniform time step $\Delta t = 5 \cdot 10^{-3}$ s.

As quantities of interest, we consider the displacement of the apex $\mathbf{A}_0$ in the vertical direction and that in the horizontal direction of a point $\mathbf{P}_0$ on the epicardium, reported in Figure 3.2, both evaluated at five time instants; thus

$$\mathbf{y}_{h,j}(\boldsymbol{\mu}) = Q_j(\mathbf{u}_h(\boldsymbol{\mu})) = \mathbf{u}_h(\mathbf{X}_{A_0}, t^{10j};\boldsymbol{\mu}) \cdot \mathbf{e}_z,$$
$$\mathbf{y}_{h,j+5}(\boldsymbol{\mu}) = Q_{j+5}(\mathbf{u}_h(\boldsymbol{\mu})) = \mathbf{u}_h(\mathbf{X}_{P_0}, t^{10j};\boldsymbol{\mu}) \cdot \mathbf{e}_x,$$

for $j = 1, \dots, \lfloor N_t/10 \rfloor$, so that $N_q = 10$. Finally, the random parameters are given by

- the material stiffness in different directions $b_f \in [6.6, 9.4]$, $b_s, b_n, b_{sn} \in [1.65, 2.35]$, $b_{fs}, b_{fn} \in [3.3, 4.7]$,
- the bulk modulus $K \in [4 \cdot 10^4, 6 \cdot 10^4]$ Pa,
- the multiplicative constant $C \in [1.5 \cdot 10^3, 2.5 \cdot 10^3]$ Pa,
- the maximum active tension $\widetilde{T}_a \in [49.5 \cdot 10^3, 70.5 \cdot 10^3]$ Pa,

- the steepness of the pressure ramp $\widetilde{p} \in [14 \cdot 10^3, 16 \cdot 10^3]$ Pa, and
- the fiber angles $\boldsymbol{\alpha}^{epi} \in [-105.5, -74.5]^\circ$ and $\boldsymbol{\alpha}^{endo} \in [74.5, 105.5]^\circ$,

so that

$$\boldsymbol{\mu} = [b_f, b_s, b_n, b_{fs}, b_{fn}, b_{sn}, K, C, \widetilde{T}_a, \widetilde{p}, \boldsymbol{\alpha}^{epi}, \boldsymbol{\alpha}^{endo}] \in \mathcal{P} \subset \mathbb{R}^{12}.$$

The Sobol' SA is performed by relying on the POD-Galerkin-DEIM ROM constructed in Section 2.3.2 with $N = 19$ and $m = 574$, and considering a base sample $N_S = 50$, thus resulting in 700 evaluations of the input-output map. We recall that is this case the FOM takes about 6 minutes for each problem solution, whereas the ROM requires 90 s, thus achieving a speed-up of 4 times compared to the high-fidelity model.

In Table 3.2 the averaged of the indices over all the quantities of interest are reported, as well as the average over the outputs related only to the apex $\mathbf{A}_0$ or the point $\mathbf{P}_0$. From the reported results we can conclude that the most influential parameters are the maximum value $\widetilde{T}_a$ of the active tension and the parameter $\boldsymbol{\alpha}^{epi}$ related to the fiber orientation at the epicardium. These results, obtained on a benchmark test for ventricular systole, can be extremely informative for the simulation of the whole cardiac cycle, especially if combined with sensitivities studies for the passive inflation test case. In fact, they allow to narrow the varying parameters to the most influential ones, in order simplify the construction of reduced order models, as detailed in the following Section 3.1.3.

Index ↓	b_f	b_s	b_n	b_{fs}	b_{fn}	b_{sn}	K	C	$\widetilde{T}_a$	$\widetilde{p}$	$\boldsymbol{\alpha}^{epi}$	$\boldsymbol{\alpha}^{endo}$
$\bar{S}_i^{tot}$	0.01	0.01	0.01	≈0	0.01	≈0	0.01	0.08	0.38	0.03	0.57	0.05
$\bar{S}_T^{tot}$	0.01	≈0	≈0	≈0	≈0	≈0	≈0	0.05	0.53	0.06	0.55	0.04
$\bar{S}_i^{\mathbf{A}_0}$	0.02	0.01	0.01	≈0	≈0	≈0	0.02	0.09	0.59	0.06	0.31	0.01
$\bar{S}_T^{\mathbf{A}_0}$	0.01	≈0	≈0	≈0	≈0	≈0	0.01	0.05	0.80	0.08	0.24	0.03
$\bar{S}_i^{\mathbf{P}_0}$	≈0	≈0	≈0	≈0	0.02	≈0	≈0	0.06	0.17	≈0	0.82	0.10
$\bar{S}_T^{\mathbf{P}_0}$	0.01	≈0	≈0	≈0	0.01	≈0	≈0	0.06	0.26	0.03	0.85	0.05

Table 3.2: Passive inflation and active contraction of a left ventricle. Sensitivity indices computed using POD-Galerkin-DEIM.

Another important aspect that arises from this analysis is that the choice of the quantities of interest affects the outcome of the SA, meaning that some factors having a great impact on a specific output may be less influential for others. To give an example for the problem at hand, the active tension has a greater impact on the contraction of the ventricle, whereas the fiber orientation controls the typical twisting motion. In view of this, relying on physics-based reduced order models, like the ones considered in this Thesis (see Chapter 2 and 4), can be extremely convenient rather than using fully data-driven emulators or surrogate models, which provide inexpensive evaluations of the input-output queries. In fact, by retaining the underlying structure of the physical model and providing an approximation at each time step of the displacement vector, physics-based ROMs allow to consider new quantities of interest without the need to repeat the whole reduction or training process.

3.1.3 Application of sensitivity analysis to reduced order models

As already mentioned at the beginning of the chapter, SA can be used to identify the less influential factors on a given output quantity, which can then be fixed at any given value in their range of variability (factor fixing), thus allowing significant model simplifications. This fact can be extremely useful also during the construction of a reduced order model. In fact, since the accuracy and the efficiency of POD-Galerkin ROMs are strictly related to the solution manifold

$$\mathcal{M}_h = \left\{ \mathbf{u}_h^n(\boldsymbol{\mu}) \in \mathbb{R}^{N_h} \mid n = 1, \dots, N_t, \ \boldsymbol{\mu} \in \mathcal{P} \right\}$$

made by all FOM solutions under variation of the parameter vector $\boldsymbol{\mu}$ over the parameter domain, considering a large number of parameters usually yields a higher-dimensional manifold, so that a larger number N of reduced basis functions would be required to accurately approximate the high-fidelity dynamics. In view of this, the following strategy can be adopted:

1. consider the parameter vector $\widehat{\boldsymbol{\mu}} \in \widehat{\mathcal{P}} \subset \mathbb{R}^{\widehat{P}}$ made of all model inputs and build the corresponding POD-Galerkin-DEIM ROM (see Chapter 2) choosing N and m in order to balance accuracy and efficiency. For instance, a sufficient (but not extremely high) level of accuracy can be first considered for this task;
2. compute Sobol' indices using Saltelli method with $\widehat{\mathbf{y}}_N(\widehat{\boldsymbol{\mu}}) = Q(\widehat{\mathbf{u}}_N(\widehat{\boldsymbol{\mu}}))$;
3. according to S_{T_i}, for $i = 1, \dots, \widehat{P}$, select the P most influential factors, that are those with higher Sobol' indices, and define the parameter vector $\boldsymbol{\mu} \in \mathcal{P} \subset \widehat{\mathcal{P}}$. All other factors can be fixed, e.g., to be equal to the midpoint of their variability range;
4. finally, build a POD-Galerkin-DEIM ROM for $\boldsymbol{\mu} \in \mathcal{P}$.

Numerical example: deformation of a Guccione clamped rectangular beam

We now apply the strategy described above to the same problem of Section 3.1.2, but now assuming that the material undergoing the deformation follows the Guccione relation (1.3), so that

$$\hat{\boldsymbol{\mu}} = [b_f, b_s, b_n, b_{fs}, b_{fn}, b_{sn}, K, C, \widetilde{p}] \in \hat{\mathcal{P}} \subset \mathbb{R}^9$$

is the vector of all model inputs for which we consider the following intervals of variability:

- $b_f \in [4, 12]$, $b_s, b_n, b_{sn} \in [1, 3]$, $b_{fs}, b_{fn} \in [2, 6]$ (material stiffness in different directions),
- $K \in [2.5 \cdot 10^4, 7.5 \cdot 10^4]$ Pa (bulk modulus),
- $C \in [1 \cdot 10^3, 3 \cdot 10^3]$ Pa (multiplicative constant), and
- $\widetilde{p} \in [2, 6]$ Pa (slope of the external load).

For the construction of the preliminary reduced model (step 1), we perform 20 FOM simulations and perform POD with tolerance $\varepsilon_{POD} = 10^{-3}$ on the snapshots matrix $\mathbf{S}_u$, thus obtaining a RB dimension of $N = 3$. Hence, the POD-Galerkin ROM is built from 50 runs of the ROM without hyper-reduction, where $m = 59$ residual basis functions are obtained choosing the POD tolerance ε_{DEIM} equal to 10^{-5}. The resulting computational speed-up with respect to the FOM is equal to 6, while the time-average $L^2(\Omega_0)$-absolute error is $\epsilon_{abs} = 6 \cdot 10^{-4}$.

Let the displacement in the vertical direction of point $\mathbf{P}_0$ (see Figure 3.1) at times $t = 0.05$ s, $t = 0.15$ s and $t = 0.25$ s be the vector of quantities of interest $\widehat{\mathbf{y}}_N(\widehat{\boldsymbol{\mu}}) = Q(\widehat{\mathbf{u}}_N(\widehat{\boldsymbol{\mu}})) \in \mathbb{R}^{N_q}$, with $N_q = 3$. The results of the SA performed using the hyper-ROM (step 2) are reported in Figure 3.3, where we have chosen $N_S = 200$, suggesting that the bulk modulus K, as well as all parameters related to the material stiffness, except for b_f, can be arbitrarily fixed (step 3).

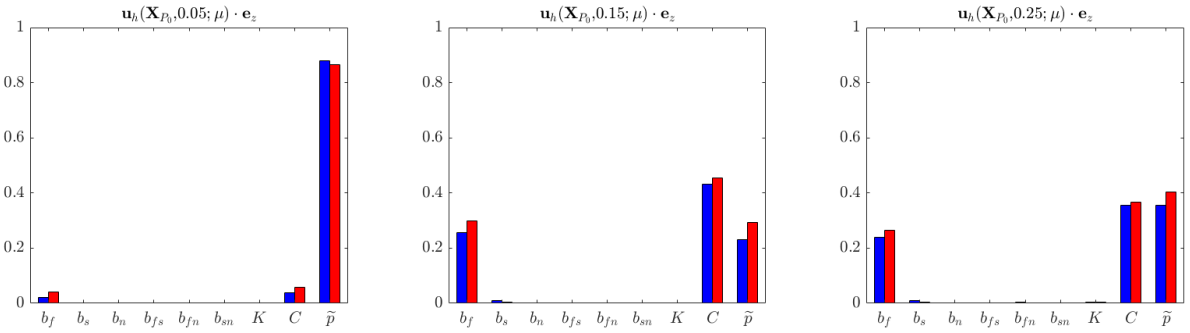


Figure 3.3: Deformation of a Guccione beam. Sensitivity indices computed using POD-Galerkin-DEIM.

This allows us to construct a simplified ROM (step 4), where the parameter vector is now given by

$$\boldsymbol{\mu} = [b_f, C, \widetilde{p}] \in \mathcal{P} \subset \mathbb{R}^3,$$

where the parameter space is $\mathcal{P} = [4, 12] \times [1 \cdot 10^3, 3 \cdot 10^3]$ Pa $\times$ $[2, 6]$ Pa. Tables 3.3 and 3.4 report the comparison between the POD-Galerkin ROMs constructed using 20 FOM simulations and the POD-Galerkin-DEIM ROMs built for $N = 3$ and 50 runs of the corresponding ROMs without hyper-reduction, respectively. Moreover, the average error and computational speed-ups of two POD-Galerkin ROMs computed over 20 testing parameters are given in Table 3.5. We can conclude that reducing the number of input parameters has an impact on both the POD dimension N and on the number of residual basis m necessary to correctly approximate the nonlinear terms. In fact, when reducing by one third the number of parameters, the required DEIM basis dimension decreases from $m = 59$ to $m = 36$, yielding to an increase of 30% of the computational speed-up. Moreover, high accuracy can be achieved for $P = 3$ using smaller values of N. This observation is crucial for the deep learning-based hyper-ROM developed in Chapter 4, since reducing both N and P yields to a less complex supervised learning task, thus allowing to rely on smaller deep neural networks.

ε_{POD}		10^{-3}	$5 \cdot 10^{-4}$	10^{-4}	$5 \cdot 10^{-5}$	10^{-5}
$\hat{P} = 9$	N	3	5	8	11	18
$P = 3$		3	4	6	7	11

Table 3.3: Deformation of a Guccione beam. RB dimensions N for POD-Galerkin ROMs.

ε_{DEIM}		$5 \cdot 10^{-4}$	10^{-4}	$5 \cdot 10^{-5}$	10^{-5}	$5 \cdot 10^{-6}$
$\hat{P} = 9$	m	46	59	66	84	94
$P = 3$		29	36	39	47	51

Table 3.4: Deformation of a Guccione beam. Residual basis dimensions m for POD-Galerkin-DEIM ROMs, with $N = 3$.

# of parameters	9	3	# of parameters	9	3
ε_{POD}	10^{-3}		Computational speed-up	$\times 6$	$\times 8$
N	3		Avg. CPU time	9 s	6.75 s
ε_{DEIM}	10^{-4}		Time-avg. $L^2(\Omega_0)$-absolute error	$6.0 \cdot 10^{-4}$	$2.3 \cdot 10^{-4}$
m	59	36	Time-avg. $L^2(\Omega_0)$-relative error	$2.4 \cdot 10^{-2}$	$1.9 \cdot 10^{-2}$

Table 3.5: Deformation of a Guccione beam. Computational data of POD-Galerkin-DEIM ROMs.

3.2 Inverse uncertainty quantification: parameter estimation

Of particular interest when dealing with parameterized models is the task of estimating input parameters based on observed data, that is, the solution to the so-called *inverse* UQ problems. Available approaches to perform inverse UQ can be grouped into two main classes: the first class consists on variational methods which provide point estimates of quantities of interest and are based on PDE-constrained optimization or filtering approaches; the second class, instead, encompasses statistical inference.

In this Thesis we focus on the second approach, particularly on Bayesian inversion, which provides a suitable framework for the estimation of parameters that are independent of time. In this setting all model inputs are described as random variables and their probability density functions (PDFs) are updated as new information is available. It follows that the goal of Bayesian inference is to improve the knowledge of the unknown inputs starting from the prior belief on their realizations and exploiting new information in order to compute the posterior PDF.

Adopting the same notation of previous section, let $(\boldsymbol{\Xi}, \mathcal{F}, \mathbb{P})$ be a probability space and $\mathbf{M} = (M_1, \dots, M_P)$ the random vector of model parameters such that

$$\begin{aligned} M_i &: \boldsymbol{\Xi} \to \mathbb{R} \\ &\xi \mapsto \mu_i = M_i(\xi), \end{aligned}$$

for $i = 1, \dots, P$. Given an observation $\mathbf{y}_{obs} \in \mathbb{R}^{N_q}$, corresponding to the realization of a random variable

$$\mathbf{Y} = y(\mathbf{M}),$$

i.e. $\mathbf{y}_{obs} = \mathbf{Y}(\xi_{obs})$, we want to infer the underlying probability distribution that produced the data. In particular, we consider $\mathbf{y}_{obs}$ to be the vector of quantities of interests evaluated on the

FOM solution $\mathbf{u}_h(\boldsymbol{\mu}^*)$, computed for a specified parameter vector $\boldsymbol{\mu}^* \in \mathcal{P}$, i.e.

$$\mathbf{y}_{obs} = Q(\mathbf{u}_h(\boldsymbol{\mu}^*)) = \mathbf{y}_h(\boldsymbol{\mu}^*),$$

where $Q\colon \mathbb{R}^{N_h \times N_t} \to \mathbb{R}^{N_q}$ is the output function whose components can be a single quantity of interest measured at different time instants or multiple measurements acquired at different locations of the computational domain, or both. Note that, with respect to the previous section, we do note assume $\mathbf{Y}$ to be a univariate model output.

In order to take into account the fact that experiments are usually affected by noise, we model the measurement error $\boldsymbol{\varepsilon} \in \mathbb{R}^{N_q}$ as additive and mutually independent of $\mathbf{M}$, so that the input-output map is actually given by

$$\mathbf{Y} = \mathbf{y}_h(\mathbf{M}) + \boldsymbol{\varepsilon},$$

and $\mathbf{y}_{obs} = \mathbf{y}_h(\boldsymbol{\mu}^*) + \boldsymbol{\varepsilon}$, where $\boldsymbol{\mu}^* = \mathbf{M}(\xi_{obs})$.

For a given data $\mathbf{y}_{obs}$, the *posterior* density of the parameter random vector $\mathbf{M}$ can be obtained using Bayes' formula as

$$\pi(\boldsymbol{\mu}|\mathbf{y}_{obs}) = \frac{\pi(\mathbf{y}_{obs}|\boldsymbol{\mu})\pi_0(\boldsymbol{\mu})}{\pi_Y(\mathbf{y}_{obs})},$$

where

- $\pi(\mathbf{y}_{obs}|\boldsymbol{\mu})$ is the *likelihood* of observing $\mathbf{y}_{obs}$ given the realization $\mathbf{M}(\xi) = \boldsymbol{\mu}$ of the input vector and represents the mechanism through which probabilities are updated;
- $\pi_0(\boldsymbol{\mu})$ is the *prior* density of the inputs and reflects any knowledge one might have about the parameters before data are acquired. Non-informative priors, such as the uniform density, can be used if no previous knowledge is available;
- $\pi_Y(\mathbf{y}_{obs})$ is the marginal density function of the output $\mathbf{Y}$ given by

  $$\pi_Y(\mathbf{y}_{obs}) = \int_{\mathcal{P}} \pi(\mathbf{y}_{obs}|\boldsymbol{\mu})\pi_0(\boldsymbol{\mu})d\boldsymbol{\mu},$$

  and is used as a normalization factor.

Thanks to the assumption of mutually independence between noise and parameters, the likelihood function is given by

$$\pi(\mathbf{y}|\boldsymbol{\mu}) = \pi_{\boldsymbol{\varepsilon}}(\mathbf{y} - y(\boldsymbol{\mu})),$$

where $\pi_{\boldsymbol{\varepsilon}}$ is the PDF of the measurement error, so that the likelihood combines the forward model and the experimental noise. Once the posterior is computed, one can repeatedly sample from its distribution in order to compute useful statistical indicators (such as the mean, the variance, the minimum and the maximum, to name a few) and to perform forward UQ, which can be run in tandem with reduced order modeling.

Algorithm 6 summarizes the Bayesian approach to inverse UQ problems for PDEs. We precise that each component of the observation vector $\mathbf{y}_{obs}$ is associated with a different quantity of interest considered. These quantities can be the same measure observed at multiple time instants and/or outputs of different nature.

Algorithm 6 Bayesian approach for inverse problems.

INPUT: $\mathbf{y}_{obs} = \mathbf{y}_h(\boldsymbol{\mu}^*) + \boldsymbol{\varepsilon} \in \mathbb{R}^{N_q}$ $(N_q \geq 1)$, $\pi_{\boldsymbol{\varepsilon}}$
OUTPUT: $\pi(\boldsymbol{\mu}|\mathbf{y}_{obs})$

1: Postulate a prior PDF $\pi_0(\boldsymbol{\mu})$ for the unknown parameter vector $\mathbf{M}$
2: Use Bayes' formula to obtain the posterior PDF

$$\pi(\boldsymbol{\mu}|\mathbf{y}_{obs}) = \frac{\pi(\mathbf{y}_{obs}|\boldsymbol{\mu})\pi_0(\boldsymbol{\mu})}{\int_{\mathcal{P}} \pi(\mathbf{y}_{obs}|\boldsymbol{\mu})\pi_0(\boldsymbol{\mu})d\boldsymbol{\mu}}$$

where $\pi(\mathbf{y}_{obs}|\boldsymbol{\mu}) = \pi_{\boldsymbol{\varepsilon}}(\mathbf{y}_{obs} - \mathbf{y}_h(\boldsymbol{\mu}))$

3.2.1 Application of reduced order models to inverse uncertainty quantification

In order to find a solution to the inverse problem, i.e. to find $\pi(\boldsymbol{\mu}|\mathbf{y}_{obs})$, we have to rely on suitable numerical techniques to explore the posterior PDF. A large class of algorithms for sampling from a probability distribution is given by the so-called Markov Chains Monte Carlo (MCMC) methods. The goal of MCMC techniques is to construct Markov chains $\{\boldsymbol{\mu}_\ell, \ell \geq 1\}$ whose stationary (target) distribution is the posterior PDF that we want to explore. We refer to, e.g., [Smi13; RC04; Sul15] for further details.

One of the most popular MCMC methods is the Metropolis-Hastings algorithm, reported in Algorithm 7 for the case at hand. The key idea is to generate a sequence of N_{MC} samples (where N_{MC} is fixed, but sufficiently large) such that, at each iteration, the new candidate value is chosen according to a specified proposal distribution π_{prop}, based on the previous sample. Hence, the new value is either accepted or rejected, according to an acceptance probability α which is determined by the properties of the likelihood and prior. In order to reduced the bias introduced by the choice of the initial sample, we perform *burn-in* by discarding $N_{burn\text{-}in}$ iterations at the beginning of an MCMC run. Finally, only one sample every N_{thin} is kept in the final chain (*thinning*), where N_{thin} is a given sub-sampling period. A common choice for the proposal π_{prop} is given by

$$\hat{\boldsymbol{\mu}}|\boldsymbol{\mu}_{\ell-1} \sim \mathcal{N}(\boldsymbol{\mu}_{\ell-1}, \sigma^2_{prop}\mathbb{I}),$$

for some suitable value of $\sigma^2_{prop} > 0$ (see [GG97]).

MCMC techniques can become computationally demanding in the context of PDE problems, as they require the repeated evaluation of the input-output map, so that relying on high-fidelity models for the solution of the underlying PDE may become unfeasible. In this Thesis we show that ROMs can be used to alleviate the computational costs in this respect, as done, e.g., in [MPL16].

Numerical example: deformation of a neo-Hookean clamped rectangular beam

Going back to the problem of the deformation of a neo-Hookean rectangular beam discussed in Section 3.1.2, we now want to perform parameter estimation using standard MCMC methods, such as the Metropolis-Hastings algorithm (see Algorithm 7). Our goal is to compare the results obtained using an expensive FOM with those computed by performing inverse UQ using reduced models. In particular, we consider the POD-Galerkin-DEIM ROM built for $N = 4$ and

Algorithm 7 Metropolis-Hasting algorithm

INPUT: N_{MC}, $N_{burn\text{-}in}$, N_{thin}, $\mathbf{y}_{obs} \in \mathbb{R}^{N_q}$, π_0, $\pi_{\boldsymbol{\varepsilon}}$, π_{prop}
OUTPUT: $\mathcal{M} = \{\boldsymbol{\mu}_\ell, \ell = 1 + N_{burn\text{-}in} : N_{thin} : N_{MC}\}$

Choose $\boldsymbol{\mu}_0 \in \mathcal{P}$ and set $\mathcal{M} = \emptyset$
for $\ell = 1, \dots, N_{MC}$ **do**
  Draw $\hat{\boldsymbol{\mu}}$ from $\pi_{prop}(\hat{\boldsymbol{\mu}}, \boldsymbol{\mu}_{\ell-1})$
  Calculate the acceptance probability

$$\alpha(\hat{\boldsymbol{\mu}}, \boldsymbol{\mu}_{\ell-1}) = \min\left(1, \frac{\pi(\mathbf{y}_{obs}|\hat{\boldsymbol{\mu}})\pi_0(\hat{\boldsymbol{\mu}})}{\pi(\mathbf{y}_{obs}|\boldsymbol{\mu}_{\ell-1})\pi_0(\boldsymbol{\mu}_{\ell-1})}\right)$$

  where $\pi(\mathbf{y}_{obs}|\,\cdot\,) = \pi_{\boldsymbol{\varepsilon}}(\mathbf{y}_{obs} - \mathbf{y}_h(\cdot))$
  Draw an independent sample from $\boldsymbol{\upsilon} \sim \mathcal{U}[0,1]$
  if $\alpha(\boldsymbol{\mu}_{\ell-1}, \hat{\boldsymbol{\mu}}) \geq \boldsymbol{\upsilon}$ **then**
    Accept $\hat{\boldsymbol{\mu}}$ and set $\boldsymbol{\mu}_\ell = \hat{\boldsymbol{\mu}}$
  else
    Reject $\hat{\boldsymbol{\mu}}$ and set $\boldsymbol{\mu}_\ell = \boldsymbol{\mu}_{\ell-1}$
  if $\ell > N_{burn\text{-}in}$ **and** $\ell \equiv 0 \bmod N_{thinning}$
    Set $\mathcal{M} \leftarrow \mathcal{M} \cup \{\boldsymbol{\mu}_\ell\}$

$m = 22$, which is characterized by a speed-up of 11 with respect to the FOM for the solution to the forward problem and a time-average $L^2(\Omega_0)$-absolute error of $\epsilon_{abs} = 7.4 \cdot 10^{-5}$.

As quantities of interest we consider the displacement in the vertical direction of $\mathbf{P}_0$ ($\mathbf{X}_{P_0} = (10^{-2}, 0.5 \cdot 10^{-3}, 0.5 \cdot 10^{-3})$) acquired every two time instants, so that

$$\mathbf{y}_{h,j}(\boldsymbol{\mu}) = Q_j(\mathbf{u}_h(\boldsymbol{\mu})) = \mathbf{u}_h(\mathbf{X}_{P_0}, t^{10j}; \boldsymbol{\mu}) \cdot \mathbf{e}_z,$$

for $j = 1, \dots, \lfloor N_t/2 \rfloor$, where $N_q = 25$.

The inputs to the Metropolis-Hastings algorithm are reported in Table 3.6. Moreover, the synthetic data $\mathbf{y}_{obs}$ are constructed by taking

$$\boldsymbol{\mu}^* = [G^*, K^*, \widetilde{p}^*] = [0.7 \cdot 10^4 \text{ Pa}, 3 \cdot 10^4 \text{ Pa}, 5 \text{ Pa}],$$

while $\boldsymbol{\mu}_0 = [G_0, K_0, \widetilde{p}_0] = [1 \cdot 10^4 \text{ Pa}, 5 \cdot 10^4 \text{ Pa}, 4 \text{ Pa}]$ is chosen as starting point for the chain.

N_{MC}	10000	π_0	$\mathcal{U}[\mathcal{P}]$ (non-informative)
$N_{burn\text{-}in}$	500	$\pi_{\boldsymbol{\varepsilon}}$	$\mathcal{N}(0, \Sigma)$, with $\Sigma = 10^{-5}\mathbb{I}$
N_{thin}	4	π_{prop}	$\mathcal{U}[\mathcal{P}]$

Table 3.6: Deformation of a neo-Hookean beam. Inputs to the Metropolis-Hastings algorithm.

The samples of the final MCMC chain are reported in Figure 3.4, where the value reported for each bin is equal to the number of elements of the bin divided by the total number of elements. In particular, the parameter value $\boldsymbol{\mu}^*$ used for generating the observed data is highlighted in red, while $\boldsymbol{\mu}_0$ corresponds to the black dotted line. These plots show that the hyper-ROM is

able to achieve the same results of the full order model, with about the same percentage of rejected candidate samples (27.5%). In particular, it turns out that the bulk modulus K is hard to identify using both models, while the shear modulus G and the steepness of the pressure ramp $\widetilde{p}$ are easier to capture. This fact confirms the results of the SA reported in Table 3.1, which identify K as the less influential parameter. For what concerns the CPU time required for the generation of the MCMC chain, the FOM requires more than 3 days of computation, while the ROM has a total run time of 7 h 40 min, showing that relying on efficient reduced models is of paramount importance to address the solution to inverse UQ problems.

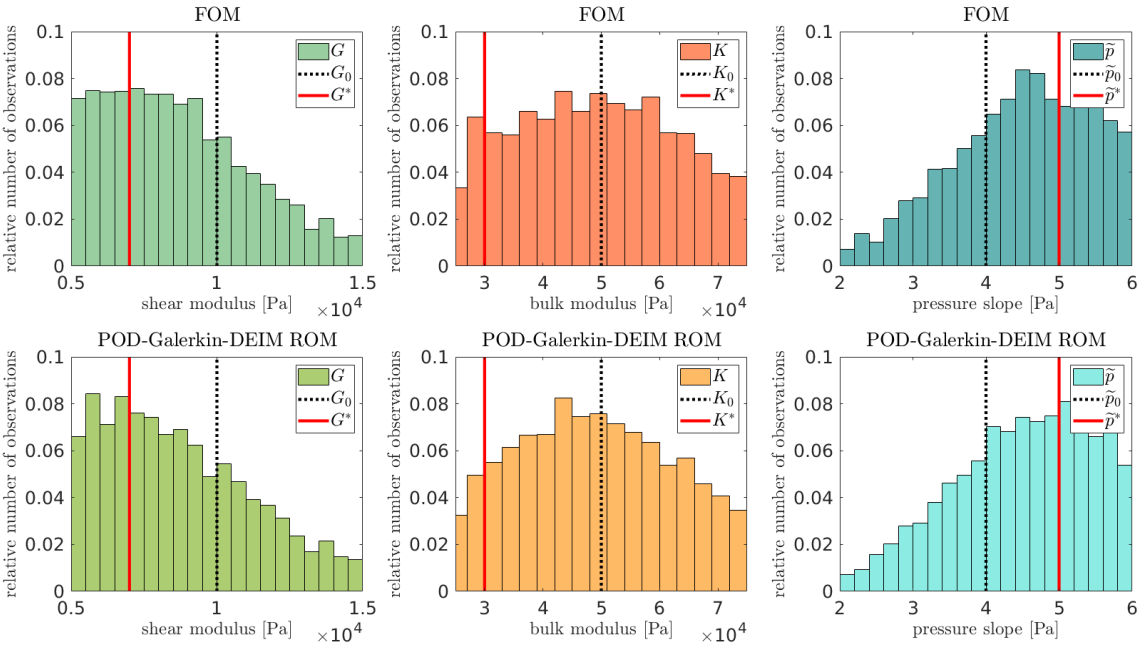


Figure 3.4: Deformation of a neo-Hookean beam. MCMC samples obtained using the FOM (top) and POD-Galerkin-DEIM ROM (bottom).

Chapter 4

Deep learning-based hyper-reduced order model network

In this chapter we develop a new deep learning-based hyper-ROM, *Deep-HyROMnet*, to overcome the computational bottleneck of POD-Galerkin-DEIM ROMs (see Chapter 2) by avoiding the assembling stage of the nonlinear arrays onto the computational mesh. First of all, we present different deep learning-based strategies to efficiently approximate the nonlinear reduced operators, based on Newton and Broyden methods; then, we describe the deep neural network (DNN) architecture employed. Finally, the new reduction approach is assessed in several scenarios and its accuracy and efficiency are compared to the classical POD-Galerkin-DEIM.

4.1 Deep learning based model order reduction

Several papers have recently introduced new methods which combine projection-based reduction with neural networks (NNs). For instance, in [Swi+19; HU18] machine learning techniques are investigated with the aim of learning the map between the input parameters and the reduced basis expansion coefficients in a non-intrusive way, while in [Hij+20] a mixed strategy based on the idea of combining projection-based methods and data-driven techniques for the solution to turbulent flows is developed. The nonlinear velocity function of a dynamical system in the ROM equations is approximated through DNNs in [GWZ20]. In order to take into account physics laws, physics-informed NNs have also been introduced [RPK19].

In the last decades, the use of machine learning techniques to perform operator inference in the context of (parameterized) differential equations has been considered, combining ideas from classical model reduction with data-driven learning. For instance, in [Lu+21] a step forward is made concerning the design of NNs able to accurately represent linear/nonlinear operators, mapping input functions to output functions. Based on the universal approximation theorem of operators [CC95], a general deep learning framework, called DeepONet, has been proposed to learn continuous operators, such as solution operators of PDEs, using DNNs. In [PW16] a non-intrusive projection-based ROM for parameterized time-dependent PDEs which are linear in the state, or include a low-order polynomial nonlinear term, is considered, and further extended to the case of non-polynomial nonlinear terms given in analytic form in [Ben+20]. In these cases, approximations of the reduced operators are inferred directly from data of the FOM by solving a least-squares problem. The advantages of projection-based model reduction and machine learning are combined in [Qia+19] to approximate linear and quadratic operators, focusing on the solution to a large class of fluid dynamics applications. Furthermore, [Bha+21] combines principal component analysis-based model reduction with a NNs for approximation, in a purely data-driven fashion, of infinite-dimensional solution maps, such as the solution

operator for time-dependent PDEs.

We recall that we seek an approximation of the nonlinear operators arising in the reduced Newton system

$$\begin{cases} \mathbf{V}^T\mathbf{J}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu})\mathbf{V}\delta\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}) = -\mathbf{V}^T\mathbf{R}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}), \\ \mathbf{u}_N^{n,(k+1)}(\boldsymbol{\mu}) = \mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}) + \delta\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}), \end{cases} \tag{4.1}$$

for $n = 1, \dots, N_t$ and $k \geq 0$, that can be computed at a low computational cost. With this aim, let $\mathbf{V} \in \mathbb{R}^{N_h \times N}$ be a N-dimensional POD basis for the ROM and define the reduced nonlinear operators as follows:

$$\begin{aligned} \mathbf{R}_N(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}) &:= \mathbf{V}^T\mathbf{R}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}) \in \mathbb{R}^N, \\ \mathbf{J}_N(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}) &:= \mathbf{V}^T\mathbf{J}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu})\mathbf{V} \in \mathbb{R}^{N \times N}, \end{aligned}$$

for $n = 1, \dots, N_t$ and $k \geq 0$. Our goal is the efficient numerical approximation of the whole sets

$$\begin{aligned} \mathcal{M}_{R_N} &= \{\mathbf{R}_N(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}) \in \mathbb{R}^N \mid n = 1, \dots, N_t,\ k \geq 0,\ \boldsymbol{\mu} \in \mathcal{P}\} \\ \mathcal{M}_{J_N} &= \{\mathbf{J}_N(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}) \in \mathbb{R}^{N \times N} \mid n = 1, \dots, N_t,\ k \geq 0,\ \boldsymbol{\mu} \in \mathcal{P}\} \end{aligned}$$

(which we refer to as *reduced residual manifold* and *reduced Jacobian manifold*, respectively) with a number of operations that depends only on the ROM dimension N and on the number of parameters P.

The main idea of our deep learning-based approach is to provide an efficient and accurate approximation of the reduced nonlinear operators arising in (4.1) by relying on DNNs. Provided the input triplets

$$\boldsymbol{\vartheta} = (\boldsymbol{\mu}, t^n, k) \in \mathcal{P} \times \{t^1, \dots, t^{N_t}\} \times \mathbb{N}^+$$

made of the input parameters $\boldsymbol{\mu} \in \mathcal{P}$, the current time step t^n, for $n = 1, \dots, N_t$, and the Newton iteration $k \geq 0$, we employ a suitable NN architecture (described in Section 4.1.1) to learn the following input-to-residual and input-to-Jacobian maps, when either the Newton method or the Broyden method are used for the solution of the ROM at each time instant:

- **Newton approach**

  $$\begin{aligned} \boldsymbol{\rho}_N &: (\boldsymbol{\mu}, t^n, k) \longmapsto \boldsymbol{\rho}_N(\boldsymbol{\mu}, t^n, k) \approx \mathbf{R}_N(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}), \\ \boldsymbol{\iota}_N &: (\boldsymbol{\mu}, t^n, k) \longmapsto \boldsymbol{\iota}_N(\boldsymbol{\mu}, t^n, k) \approx \mathbf{J}_N(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}), \end{aligned}$$

  for the reduced residual and the reduced Jacobian, respectively. The linear system in (4.1) is finally replaced by

  $$\boldsymbol{\iota}_N(\boldsymbol{\mu}, t^n, k)\delta\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}) = -\boldsymbol{\rho}_N(\boldsymbol{\mu}, t^n, k). \tag{4.2}$$

  We refer to this approach as the *Deep-HyROMnet* method.

- **Broyden approach**

  $$\boldsymbol{\rho}_N : (\boldsymbol{\mu}, t^n, k) \longmapsto \boldsymbol{\rho}_N(\boldsymbol{\mu}, t^n, k) \approx \mathbf{R}_N(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}),$$

for the reduced residual. For $k \geq 0$, the reduced Jacobian $\mathbf{J}_N(\mathbf{Vu}_N^{n,(k+1)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu})$ can be approximated as

$$\widetilde{\mathbf{J}}_N(\star^{(k+1)}) = \widetilde{\mathbf{J}}_N(\star^{(k)}) + \frac{\Delta\boldsymbol{\rho}_N(\boldsymbol{\mu}, t^n, k+1) - \widetilde{\mathbf{J}}_N(\star^{(k)})\delta\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu})}{(\delta\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}))^T(\delta\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}))}(\delta\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}))^T,$$

where we denoted $(\star^{(j)}) = (\mathbf{Vu}_N^{n,(j)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu})$, for $j \in \{k, k+1\}$, and defined the residual increment as $\Delta\boldsymbol{\rho}_N(\boldsymbol{\mu}, t^n, k+1) = \boldsymbol{\rho}_N(\boldsymbol{\mu}, t^n, k+1) - \boldsymbol{\rho}_N(\boldsymbol{\mu}, t^n, k)$. The linear system in (4.1) is in this case replaced by

$$\widetilde{\mathbf{J}}_N(\mathbf{Vu}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu})\delta\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}) = -\boldsymbol{\rho}_N(\boldsymbol{\mu}, t^n, k). \tag{4.3}$$

In order to initialize $\widetilde{\mathbf{J}}_N(\mathbf{Vu}_N^{n,(0)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu})$, different strategies are possible, such as:

1. learn the input-to-Jacobian map

$$\boldsymbol{\iota}_N \colon (\boldsymbol{\mu}, t^n, k) \longmapsto \boldsymbol{\iota}_N(\boldsymbol{\mu}, t^n, k) \approx \mathbf{J}_N(\mathbf{Vu}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}),$$

   for all $k \geq 0$ and set $\widetilde{\mathbf{J}}_N(\mathbf{Vu}_N^{n,(0)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}) = \boldsymbol{\iota}_N(\boldsymbol{\mu}, t^n, 0)$;

2. learn the input-to-Jacobian map

$$\boldsymbol{\iota}_{N,0} \colon (\boldsymbol{\mu}, t^n) \longmapsto \boldsymbol{\iota}_{N,0}(\boldsymbol{\mu}, t^n) \approx \mathbf{J}_N(\mathbf{Vu}_N^{n,(0)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu})$$

   and set $\widetilde{\mathbf{J}}_N(\mathbf{Vu}_N^{n,(0)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}) = \boldsymbol{\iota}_{N,0}(\boldsymbol{\mu}, t)$;

3. given a DEIM basis $\boldsymbol{\Phi}_{\mathcal{R}} \in \mathbb{R}^{N_h \times m}$, set

$$\widetilde{\mathbf{J}}_N(\mathbf{Vu}_N^{n,(0)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}) = \mathbf{V}^T\boldsymbol{\Phi}_{\mathcal{R}}({\boldsymbol{\Phi}_{\mathcal{R}}}^{-1})_{|\mathcal{I}}\mathbf{J}(\mathbf{Vu}_N^{n,(0)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu})_{|\mathcal{I}}\mathbf{V}.$$

This procedure guarantees to recover the effectiveness of the offline-online decoupling characteristic of the RB method (see Section 2.1.1). During the offline stage, we need to collect FOM solution snapshots for the construction of the reduced basis $\mathbf{V}$, perform POD-Galerkin ROM simulations for a new set of parameters (as done for the DEIM-based approach reported in Section 2.2) to collect the reduced nonlinear operators and then, train the DNNs on these data. Online, for each new instance of the parameter, the outputs of the networks are evaluated in order to assemble efficiently the reduced Newton system and the N-dimensional ROM is solved rapidly. We point out that the inputs given to the DNNs to perform supervised learning are low-dimensional arrays, so that the overwhelming training times and costs that may be required by even moderately large high-fidelity dimensions can be avoided.

To summarize, in the case of the Newton approach, we end up with the following reduced problem: given $\boldsymbol{\mu} \in \mathcal{P}$ and, for $n = 1, \ldots, N_t$, the initial guess $\mathbf{u}_N^{n,(0)}(\boldsymbol{\mu}) = \mathbf{u}_N^{n-1}(\boldsymbol{\mu})$, find $\delta\mathbf{u}_N^{n,(k)} \in \mathbb{R}^N$ such that, for $k \geq 0$,

$$\begin{cases} \boldsymbol{\iota}_N(\boldsymbol{\mu}, t^n, k)\delta\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}) = -\boldsymbol{\rho}_N(\boldsymbol{\mu}, t^n, k), \\ \mathbf{u}_N^{n,(k+1)}(\boldsymbol{\mu}) = \mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}) + \delta\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}), \end{cases} \tag{4.4}$$

until $\|\boldsymbol{\rho}_N(\boldsymbol{\mu}, t^n, k)\|_2 / \|\boldsymbol{\rho}_N(\boldsymbol{\mu}, t^n, 0)\|_2 < \varepsilon_{Nwt}$, where $\varepsilon_{Nwt} > 0$ is a given tolerance. It is important to point out that, except for the approximation error of the reduced nonlinear operators, the Deep-HyROMnet approach described in 4.1 is a physics-based method and that the computed displacement solves a slightly modified nonlinear elastodynamics equation, because of the approximation of the ROM arrays.

In Algorithm 8, we report a summary of both the offline stage and the online stage of the hyper-ROM Deep-HyROMnet, based on Newton method (4.2).

Algorithm 8 Deep-HyROMnet for nonlinear time-dependent problems

Offline stage

INPUT: $\boldsymbol{\mu}_\ell$, for $\ell = 1, \dots, n_s$, and $\boldsymbol{\mu}_{\ell'}$, for $\ell' = 1, \dots, n'_s$

OUTPUT: $\mathbf{V} \in \mathbb{R}^{N_h \times N}$

for $\ell = 1, \dots, n_s$ **do**
  for $n = 1, \dots, N_t$ **do**
    for $k \geq 0$ **until convergence do**
      Assemble and solve problem (1.13)
      Collect $\mathbf{S}_u \leftarrow \mathbf{S}_u \cup \left[\mathbf{u}_h^{n,(k)}(\boldsymbol{\mu}_\ell)\right]$ column-wise
Construct $\mathbf{V} = \text{POD}(\mathbf{S}_u, \varepsilon_{POD})$ (see Algorithm 2)
for $\ell' = 1, \dots, n'_s$ **do**
  for $n = 1, \dots, N_t$ **do**
    for $k \geq 0$ **until convergence do**
      Assemble and solve reduced problem (4.1)
      Collect $\mathbf{S}_{\boldsymbol{\rho}} \leftarrow \mathbf{S}_{\boldsymbol{\rho}} \cup \left[\mathbf{V}^T \mathbf{R}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}_{\ell'}), t^n; \boldsymbol{\mu}_{\ell'})\right]$ column-wise
      Collect $\mathbf{S}_{\boldsymbol{\iota}} \leftarrow \mathbf{S}_{\boldsymbol{\iota}} \cup \left[\mathbf{V}^T \mathbf{J}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}_{\ell'}), t^n; \boldsymbol{\mu}_{\ell'})\mathbf{V}\right]$ column-wise
Train the DNNs (see Algorithm 9)

Online stage

INPUT: $\boldsymbol{\mu} \in \mathcal{P}$

OUTPUT: $\mathbf{V}\mathbf{u}_N^n(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$, for $n = 1, \dots, N_t$

for $n = 0, \dots, N_t - 1$ **do**
  for $k \geq 0$ **until convergence do**
    Compute $\boldsymbol{\rho}_N(\boldsymbol{\mu}, t^n, k)$ and $\boldsymbol{\iota}_N(\boldsymbol{\mu}, t^n, k)$ (see Algorithm 10)
    Solve hyper-reduced problem (4.4)
Recover $\mathbf{V}\mathbf{u}_N^n(\boldsymbol{\mu})$, for $n = 1, \dots, N_t$

4.1.1 DL-ROM-based neural network

For the sake of generality, we will focus on the DNN-based approximation of the reduced residual vector only, that is

$$\boldsymbol{\rho}_N(\boldsymbol{\mu}, t^n, k) \approx \mathbf{R}_N(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}) \in \mathbb{R}^N.$$

In fact, by relying on a suitable transformation, we can easily write the Jacobian matrix as a vector of dimension N^2 and apply the same procedure described in the following for the residual

vector also in the case of the Jacobian matrix. In particular, we define the transformation

$$vec\colon \mathbb{R}^{N\times N} \to \mathbb{R}^{N^2}, \quad vec(\mathbf{J}_N(\mathbf{Vu}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu})) = \mathbf{j}_N(\mathbf{Vu}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}),$$

which consists in stacking the columns of $\mathbf{J}_N(\mathbf{Vu}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu})$ in a vector on which is then applied the DNN, thus obtaining

$$\widetilde{\boldsymbol{\iota}}_N(\boldsymbol{\mu}, t^n, k) \approx \mathbf{j}_N(\mathbf{Vu}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}) \in \mathbb{R}^{N^2}.$$

Finally, we revert the *vec* operation, so that $\boldsymbol{\iota}_N(\boldsymbol{\mu}, t^n, k) = vec^{-1}(\widetilde{\boldsymbol{\iota}}_N(\boldsymbol{\mu}, t^n, k))$.

Our aim is to efficiently approximate the whole set $\mathcal{M}_{R_N}$ by means of the reduced residual trial manifold, defined as

$$\mathcal{M}_{\rho_N} = \{\boldsymbol{\rho}_N(\boldsymbol{\mu}, t^n, k) \mid \boldsymbol{\mu} \in \mathcal{P},\ n = 1, \dots, N_t,\ k \geq 0\} \subset \mathbb{R}^N.$$

The DL-ROM-based [FDM21] approximation of the ROM residual $\mathbf{R}_N(\mathbf{Vu}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu})$ takes the form

$$\boldsymbol{\rho}_N(\boldsymbol{\mu}, t^n, k) = \widetilde{\mathbf{R}}_N(\boldsymbol{\mu}, t^n, k; \boldsymbol{\theta}_{DF}, \boldsymbol{\theta}_D) = \mathbf{f}_N^D(\boldsymbol{\phi}_q^{DF}(\boldsymbol{\mu}, t^n, k; \boldsymbol{\theta}_{DF}); \boldsymbol{\theta}_D)$$

where

- $\boldsymbol{\phi}_q^{DF}(\cdot\,; \boldsymbol{\theta}_{DF})\colon \mathbb{R}^{P+2} \to \mathbb{R}^q$ such that

  $$\boldsymbol{\phi}_q^{DF}(\boldsymbol{\mu}, t^n, k; \boldsymbol{\theta}_{DF}) = \mathbf{R}_q(\boldsymbol{\mu}, t^n, k; \boldsymbol{\theta}_{DF})$$

  is a deep feedforward neural network (DFNN), consisting in the subsequent composition of a nonlinear activation function, applied to a linear transformation of the input, multiple times. Here, $\boldsymbol{\theta}_{DF}$ denotes the vector of parameters of the DFNN, collecting all the corresponding weights and biases of each layer and q is as close as possible to the input size $P + 2$;

- $\mathbf{f}_N^D(\cdot\,; \boldsymbol{\theta}_D)\colon \mathbb{R}^q \to \mathbb{R}^N$ such that

  $$\mathbf{f}_N^D(\mathbf{R}_q(\boldsymbol{\mu}, t^n, k; \boldsymbol{\theta}_{DF}); \boldsymbol{\theta}_D) = \widetilde{\mathbf{R}}_N(\boldsymbol{\mu}, t^n, k; \boldsymbol{\theta}_{DF}, \boldsymbol{\theta}_D)$$

  is the decoder function of a convolutional autoencoder (CAE), obtained as the composition of several layers (some of which are convolutional), depending upon a vector $\boldsymbol{\theta}_D$ collecting all the corresponding weights and biases.

The encoder function of the CAE is exploited, during the training stage only, to map the reduced residual $\mathbf{R}_N(\mathbf{Vu}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu})$ associated to $(\boldsymbol{\mu}, t^n, k)$ onto a low-dimensional representation

$$\mathbf{f}_q^E(\mathbf{R}_N(\mathbf{Vu}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}); \boldsymbol{\theta}_E) = \widetilde{\mathbf{R}}_q(\boldsymbol{\mu}, t^n, k; \boldsymbol{\theta}_E),$$

where $\mathbf{f}_q^E(\cdot\,; \boldsymbol{\theta}_E)\colon \mathbb{R}^N \to \mathbb{R}^q$ denotes the encoder function, depending upon a vector $\boldsymbol{\theta}_E$ of parameters.

Remark 4.1

We point out that the input of the encoder function, that is, the reduced residual vector $\mathbf{R}_N$, is reshaped into a square matrix by rewriting its elements in row-major order, thus obtaining $\mathbf{R}_N^{reshape} \in \mathbb{R}^{\sqrt{N}\times\sqrt{N}}$. If N is not a square, the input $\mathbf{R}_N$ is zero-padded as explained in [GBC16], and the additional elements are subsequently discarded.

Regarding the prediction of the reduced residual for a new unseen instance of the input $\boldsymbol{\mu}_{test} \in \mathcal{P}$, computing the DL-ROM-based approximation of $\mathbf{R}_N(\mathbf{Vu}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}_{test})$, for any possible $n = 1, \ldots, N_t$ and $k \geq 0$, corresponds to the testing stage of a DFNN and of the decoder function of a convolutional AE; meaning that, at testing time, we discard the encoder function. The architecture used during training is reported in Figure 4.1, whereas, during the testing phase, the encoder function is discarded.

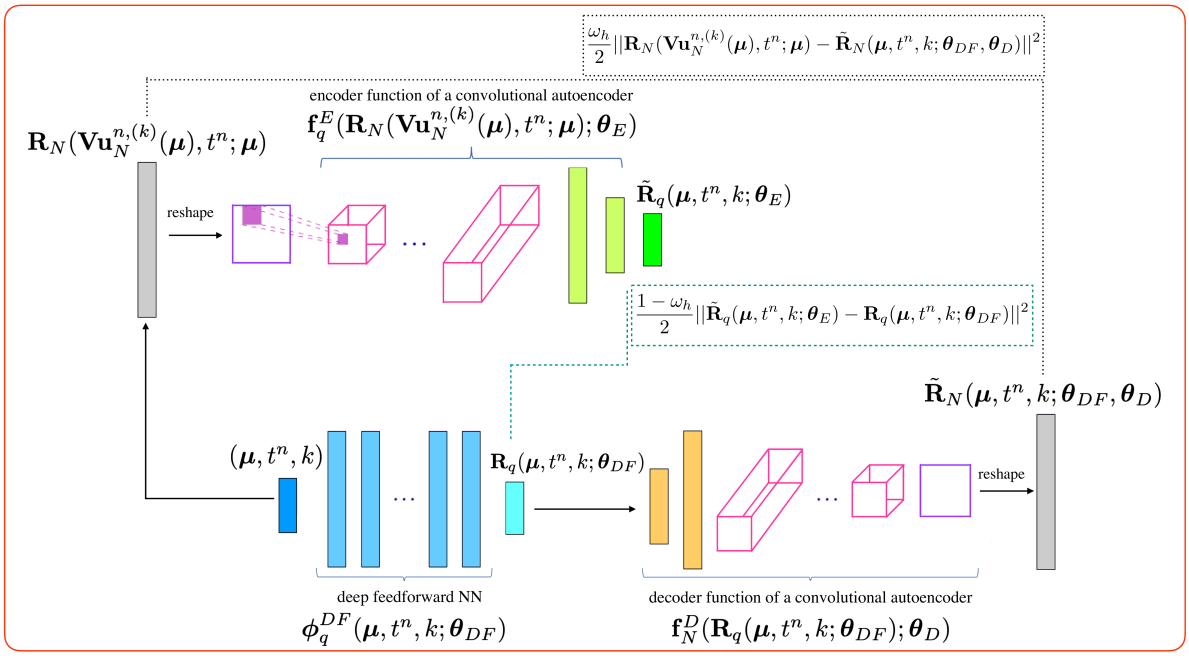


Figure 4.1: DNN architecture used during the training phase for the reduced residual vector.

Let us define the reduced residual snapshots matrix $\mathbf{S}_{\boldsymbol{\rho}} \in \mathbb{R}^{N \times N_{train}}$, with $N_{train} = n_s' N_t N_k$, as

$$\mathbf{S}_{\boldsymbol{\rho}} = \left[\mathbf{R}_N(\mathbf{Vu}_N^{n,(k)}(\boldsymbol{\mu}_\ell), t^n; \boldsymbol{\mu}_\ell), \;\; k \geq 0\right]_{n=1,\ldots,N_t}^{\ell=1,\ldots,n_s'},$$

that is, the matrix collecting column-wise ROM residuals computed for n_s' sampled parameters $\boldsymbol{\mu}_\ell \in \mathcal{P}$, at different time instances $t^1, \ldots, t^{N_t}$ and for each Newton iteration $k \geq 0$, and the parameter matrix $\mathbf{M} \in \mathbb{R}^{(P+2) \times N_{train}}$ of the corresponding triples

$$\mathbf{M} = \left[(\boldsymbol{\mu}_\ell, t^n, k), \;\; k \geq 0\right]_{n=1,\ldots,N_t}^{\ell=1,\ldots,n_s'}.$$

The training stage consists in solving the following optimization problem in the weights variable $\boldsymbol{\theta} = (\boldsymbol{\theta}_E, \boldsymbol{\theta}_{DF}, \boldsymbol{\theta}_D)$:

$$\min_{\boldsymbol{\theta}} \mathcal{J}(\boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} \frac{1}{N_{train}} \sum_{\ell=1}^{n_s'} \sum_{n=1}^{N_t} \sum_{k=0}^{N_k} \mathcal{L}(\boldsymbol{\mu}_\ell, t^n, k; \boldsymbol{\theta}),$$

where

$$\begin{aligned} \mathcal{L}(\boldsymbol{\mu}_\ell, t^n, k; \boldsymbol{\theta}) = & \frac{\omega_h}{2} \|\mathbf{R}_N(\mathbf{Vu}_N^{n,(k)}(\boldsymbol{\mu}_\ell), t^n; \boldsymbol{\mu}_\ell) - \tilde{\mathbf{R}}_N(\boldsymbol{\mu}_\ell, t^n, k; \boldsymbol{\theta}_{DF}, \boldsymbol{\theta}_D)\|^2 \\ & + \frac{1-\omega_h}{2} \|\tilde{\mathbf{R}}_q(\boldsymbol{\mu}_\ell, t^n, k; \boldsymbol{\theta}_E) - \mathbf{R}_q(\boldsymbol{\mu}_\ell, t^n, k; \boldsymbol{\theta}_{DF})\|^2, \end{aligned} \tag{4.5}$$

with $\omega_h \in [0,1]$. The loss function (4.5) combines the reconstruction error, i.e. the error between the ROM residual and the DL-ROM-based approximation, and the error between the intrinsic coordinates and the output of the encoder. The training stage of the DNNs involved in Deep-HyROMnet is detailed in Algorithm 9; in particular, we denote by α the training-validation splitting fraction, by η the starting learning rate, by N_b the batch size, by $n_b = (1-\alpha)N_{train}/N_b$ the number of minibatches and by N_e the maximum number of epochs. The testing stage of the DNNs is detailed in Algorithm 10. See, e.g., [FDM21; FM22] for further details.

Algorithm 9 Training stage for the DNNs, based on Algorithm 1 of [FDM21]

INPUT: $\mathbf{M} \in \mathbb{R}^{(P+2)\times N_{train}}$, $\mathbf{S} \in \mathbb{R}^{N\times N_{train}}$ (i.e. $\mathbf{S}_{\boldsymbol{\rho}}$ or $\mathbf{S}_{\boldsymbol{\iota}}$), α, η, N_b, n_b, N_e, early-stopping
OUTPUT: $\boldsymbol{\theta}^* = (\boldsymbol{\theta}^*_E, \boldsymbol{\theta}^*_{DF}, \boldsymbol{\theta}^*_D)$ (optimal)

1: Randomly shuffle $\mathbf{M}$ and $\mathbf{S}$
2: Split data in $\mathbf{M} = \left[\mathbf{M}^{train}, \mathbf{M}^{val}\right]$ and $\mathbf{S} = \left[\mathbf{S}^{train}, \mathbf{S}^{val}\right]$ (according to α)
3: Normalize $\mathbf{M}$ and $\mathbf{S}$ according to (4.6)
4: Randomly initialize $\boldsymbol{\theta}^0 = (\boldsymbol{\theta}^0_E, \boldsymbol{\theta}^0_{DF}, \boldsymbol{\theta}^0_D)$
5: $n_e = 0$
6: **while** ¬early-stopping **and** $n_e \leq N_e$ **do**
7: **for** $k = 1, \dots, n_b$ **do**
8: Sample a minibatch $(\mathbf{M}^{batch}, \mathbf{S}^{batch}) \subset (\mathbf{M}^{train}, \mathbf{S}^{train})$
9: $\mathbf{S}^{batch} = reshape(\mathbf{S}^{batch})$
10: $\widetilde{\mathbf{S}}^{batch}_q(\boldsymbol{\theta}^{n_b n_e+k}_E) = \mathbf{f}^E_q(\mathbf{S}^{batch}; \boldsymbol{\theta}^{n_b n_e+k}_E)$
11: $\mathbf{S}^{batch}_q(\boldsymbol{\theta}^{n_b n_e+k}_{DF}) = \boldsymbol{\phi}^{DF}_q(\mathbf{M}^{batch}; \boldsymbol{\theta}^{n_b n_e+k}_{DF})$
12: $\widetilde{\mathbf{S}}^{batch}_N(\boldsymbol{\theta}^{n_b n_e+k}_{DF}, \boldsymbol{\theta}^{n_b n_e+k}_D) = \mathbf{f}^D_N(\mathbf{S}^{batch}_q(\boldsymbol{\theta}^{n_b n_e+k}_{DF}); \boldsymbol{\theta}^{n_b n_e+k}_D)$
13: $\widetilde{\mathbf{S}}^{batch}_N = reshape(\widetilde{\mathbf{S}}^{batch}_N)$
14: Accumulate loss (4.5) on $(\mathbf{M}^{batch}, \mathbf{S}^{batch})$ and compute $\hat{\nabla}_{\boldsymbol{\theta}}\mathcal{J}$
15: $\boldsymbol{\theta}^{n_b n_e+k+1} = \text{ADAM}(\eta, \hat{\nabla}_{\boldsymbol{\theta}}\mathcal{J}, \boldsymbol{\theta}^{n_b n_e+k})$
16: Repeat instructions 9-13 on $(\mathbf{M}^{val}, \mathbf{S}^{val})$ to evaluate early-stopping criterion
17: $n_e = n_e + 1$

Algorithm 10 Testing stage for the DNNs, based on Algorithm 2 of [FDM21]

INPUT: $(\boldsymbol{\mu}, t^n, k) \in \mathcal{P} \times \{t^1, \dots, t^{N_t}\} \times \mathbb{N}^+$, $(\boldsymbol{\theta}^*_{DF}, \boldsymbol{\theta}^*_D)$ (optimal)
OUTPUT: $\widetilde{\mathbf{S}}_N$ (i.e. $\boldsymbol{\rho}_N(\boldsymbol{\mu}, t^n, k)$ or $\boldsymbol{\iota}_N(\boldsymbol{\mu}, t^n, k)$)

1: $\mathbf{S}_q(\boldsymbol{\theta}^*_{DF}) = \boldsymbol{\phi}^{DF}_q(\boldsymbol{\mu}, t^n, k; \boldsymbol{\theta}^*_{DF})$
2: $\widetilde{\mathbf{S}}_N(\boldsymbol{\theta}^*_{DF}, \boldsymbol{\theta}^*_D) = \mathbf{f}^D_N(\mathbf{S}_q(\boldsymbol{\theta}^*_{DF}); \boldsymbol{\theta}^*_D)$
3: $\widetilde{\mathbf{S}}_N = reshape(\widetilde{\mathbf{S}}_N)$

Remark 4.2
Differently from the scaling techniques used in [FDM21; FM22], which are based on a min-max procedure, we standardize the input and output of the DNN as follows. After splitting the data into training and validation sets according to a user-defined training-validation splitting fraction, $\mathbf{M} = \left[\mathbf{M}^{train}, \mathbf{M}^{val}\right]$ and $\mathbf{S} = \left[\mathbf{S}^{train}, \mathbf{S}^{val}\right]$, we define for each row of the training set

the corresponding mean and standard deviation

$$M^i_{mean} = \frac{1}{N_{train}} \sum_{j=1}^{N_{train}} M^{train}_{ij}, \quad M^i_{sd} = \sqrt{\frac{1}{N_{train}-1} \sum_{j=1}^{N_{train}} (M^{train}_{ij} - M^i_{mean})^2},$$

so that parameters are normalized by applying the following transformation

$$M^{train}_{ij} \mapsto \frac{M^{train}_{ij} - M^i_{mean}}{M^i_{sd}}, \quad i = 1, \dots, P+2, \quad j = 1, \dots, N_{train} \tag{4.6}$$

that is, each feature of the training parameter matrix is standardized. The same procedure is applied to the training snapshots matrix $\mathbf{S}^{train}$ by replacing M^i_* with S^i_*, where $* \in \{mean, sd\}$. Transformation (4.6) is applied to the validation and testing sets as well, but considering the mean and the standard deviation computed over the training set. In order to rescale the reconstructed solution to the original values, we apply the inverse transformation.

Remark 4.3
We have observed (see Table 4.1) that the computational time required by the GPU to compute the outputs of the DNNs does not scale linearly with respect to the number of input instances (at least when this number is smaller than the number of cores, in our case $\mathcal{O}(10^4)$). This means that computing the reduced operators through the DNNs for a single instance of the input triple $\boldsymbol{\vartheta} = (\boldsymbol{\mu}, t^n, k)$, when n and k are fixed, requires almost the same time necessary to compute the same quantities for all possible values of $n = 1, \dots, N_t$ and $k = 0, \dots, N_{nwt}$, being N_{nwt} an upper bound for the number of Newton iterations to be used at each time step. Because of this, in our numerical examples we pre-compute the DL-ROM-based approximations of the reduced residual vectors and Jacobian matrices for a given parameter $\boldsymbol{\mu} \in \mathcal{P}$, and read them from file when solving the reduced systems (4.2). In particular, we consider $N_{nwt} = N^{FOM}_{nwt} + 3$, where N^{FOM}_{nwt} is the maximum number of Newton iterations required by the FOM during the offline stage. We point out that the computational time required for this stage is taken into account into the total online time.

Test case	4.2.1	4.2.1	4.2.1	4.2.2	4.2.2	5.3.1	5.3.2
N	4	4	4	16	16	63	80
# of instances	1					–	–
GPU time for residuals	1.1 ms	1.1 ms	1.1 ms	1.2 ms	1.2 ms	–	–
GPU time for Jacobians	1.2 ms	1.2 ms	1.2 ms	1.3 ms	1.3 ms	–	–
# of instances	300	420	450	300	300	1920	1920
GPU time for residuals	1.7 ms	1.9 ms	2 ms	1.8 ms	1.8 ms	8.1 ms	8.3 ms
GPU time for Jacobians	1.8 ms	2 ms	2.2 ms	2.6 ms	2.6 ms	70 ms	180 ms

Table 4.1: Computational times required to compute a given number of instances of the DNN-outputs, for different test cases.

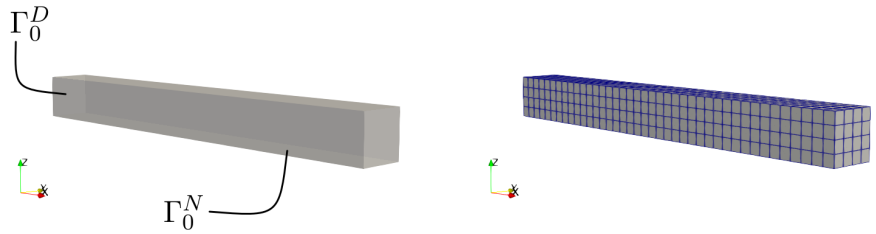


Figure 4.2: Rectangular beam geometry (left) and computational grid (right).

4.2 Numerical results

In this section we measure the performances of the proposed reduction strategy on the numerical examples considered in Section 2.3, specifically,

(i) a series of structural tests on a simple rectangular geometry with different loading boundary conditions;

(ii) two benchmark tests on a prolate spheroid geometry simulating cardiac relaxation and contraction,

and compare the results with those obtained when using POD-Galerkin-DEIM ROMs.

We recall that, as a measure of the accuracy of the ROM with respect to the FOM, for a given parameter instance, we consider time-averaged L^2-errors of the displacement vector, that are

$$\epsilon_{abs}(\boldsymbol{\mu}) = \frac{1}{N_t}\sum_{n=1}^{N_t}\|\mathbf{u}_h(\cdot,t^n;\boldsymbol{\mu}) - \mathbf{V}\mathbf{u}_N(\cdot,t^n;\boldsymbol{\mu})\|_2$$
$$\epsilon_{rel}(\boldsymbol{\mu}) = \frac{1}{N_t}\sum_{n=1}^{N_t}\frac{\|\mathbf{u}_h(\cdot,t^n;\boldsymbol{\mu}) - \mathbf{V}\mathbf{u}_N(\cdot,t^n;\boldsymbol{\mu})\|_2}{\|\mathbf{u}_h(\cdot,t^n;\boldsymbol{\mu})\|_2}.$$

Moreover, the ratio between FOM and ROM online computational times is used to measure the ROM efficiency, since it represents the speed-up of the ROM with respect to the FOM. All the computations have been performed on a PC desktop computer with 3.70GHz Intel Core i5-9600K CPU and 16GB RAM.

4.2.1 Deformation of a clamped rectangular beam

The first series of test cases represents a typical structural mechanics problem, with reference geometry $\bar{\Omega}_0 = [0, 10^{-2}] \times [0, 10^{-3}] \times [0, 10^{-3}]$ m, reported in Figure 4.2. The beam is clamped at the left-hand side, that is, Dirichlet boundary conditions are imposed on the left face $x = 0$ m, whilst a pressure load changing with the deformed surface orientation is applied to the entire bottom face $z = 0$ m (i.e. Γ_0^N). Homogeneous Neumann conditions are applied on the remaining boundaries. For the cases at hand, we consider a nearly-incompressible neo-Hookean material, yielding the following first Piola-Kirchhoff stress tensor, characterized by a non-polynomial nonlinearity,

$$\mathbf{P}(\mathbf{F}) = GJ^{-\frac{2}{3}}\left(\mathbf{F} - \frac{1}{3}\mathcal{I}_1\mathbf{F}^T\right) + \frac{K}{2}J\left(J - 1 + \frac{1}{J}\ln(J)\right)\mathbf{F}^T.$$

Here $G > 0$ denotes the shear modulus, $\mathcal{I}_1 = J^{-\frac{2}{3}} \det(\mathbf{C})$ and $K > 0$ is the bulk modulus penalizing volume variations. The FOM is built on a hexahedral mesh with 640 elements and 1025 vertices and $\mathbb{Q}_1$-FE are employed, so that the high-fidelity dimension is equal to $N_h = 3075$.

Test case 1: linear function for the pressure load

As first test case, we consider a situation in which the beam is progressively loaded by choosing as pressure boundary condition the following parameterized linear function,

$$\mathbf{g}(t; \boldsymbol{\mu}) = \widetilde{p}\, t/T, \quad t \in (0, T).$$

We choose $T = 0.25$ s as final time instance and employ a uniform time step $\Delta t = 5 \cdot 10^{-3}$ s for the time discretization scheme, resulting in a total number of 50 time steps. As parameters, we consider:

- the shear modulus $G \in [0.5 \cdot 10^4, 1.5 \cdot 10^4]$ Pa;
- the bulk modulus $K \in [2.5 \cdot 10^4, 7.5 \cdot 10^4]$ Pa;
- the external load parameter $\widetilde{p} \in [2, 6]$ Pa.

For the construction of the hyper-ROM, we consider a reduced basis $\mathbf{V} \in \mathbb{R}^{N_h \times N}$ build by performing POD on the solution snapshots matrix $\mathbf{S}_u$ for $n_s = 50$. In this case we consider $N = 4$, which yields a projection error (that is the error of the POD-Galerkin ROM without hyper-reduction) of $\epsilon_{rel} \approx 10^{-2}$. As observed in Section 2.3, POD-Galerkin ROMs achieve good accuracy even in presence of a handful of basis functions, so that choosing small values of N facilitates the training of the DNNs which are used to approximate low-dimensional structures. Finally, we perform $n_s' = 200$ ROM simulation in order to collect reduced-residual and reduced-Jacobian data for training the DNNs (see Algorithm 8).

In Table 4.2 we report the computational data related to the DEIM-based hyper-ROMs (see Table 2.1 for further details), which clearly shows that the computational bottleneck of the POD-Galerkin-DEIM technique is the construction of the reduced system at each Newton

DEIM interpolation dofs m	25	33	43
Online CPU time	2.8 s	3.6 s	4.3 s
◦ system construction [∗]	78%	83%	88%
◦ system solution	0.16%	0.13%	0.09%
[∗] System construction for each Newton iteration	0.02 s	0.02 s	0.03 s
◦ residual assembling	89%	87%	88%
◦ Jacobian computing through AD	0.6%	0.4%	0.5%

Table 4.2: Test case 1. Computational data related to POD-Galerkin-DEIM with $N = 4$ and different values of m.

iteration, which requires between 78% and 88% of the total CPU time. In particular, almost 90% of this computational time is demanded for assembling the residual $\mathbf{R}(\mathbf{Vu}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu})$ on the reduced mesh, while computing the associated Jacobian matrix using the automatic differentiation tool takes less than 1%.

To assess the performances of the deep learning-based hyper-ROMs (4.2) and (4.3), and compare their efficiency and accuracy with respect to classical hyper-reduction approaches, we consider the POD-Galerkin-DEIM ROMs obtained for a given DEIM residual basis $\boldsymbol{\Phi}_{\mathcal{R}} \in \mathbb{R}^{N_h \times m}$ of dimension $m = 22$ (DEIM-22) and $m = 30$ (DEIM-30), respectively, which represent good trade-offs between accuracy and computational speed-up. Hence, we compute the corresponding reduced solutions on a testing set of 50 instances of the parameter vector, different form the ones used during the offline stage. The computational speed-up with respect to the FOM, the absolute error ϵ_{abs} and the relative error ϵ_{rel}, averaged over the testing set, are reported in Table 4.3.

	DEIM-22	DEIM-30	Deep-HyROMnet	Broyden 1.	Broyden 2.
Speed-up	×11	×8	×1012	×987	×978
Avg. CPU time	2 s	3 s	0.026 s	0.026 s	0.027 s
$\text{mean}_{\boldsymbol{\mu}}\ \epsilon_{abs}(\boldsymbol{\mu})$	$7.4 \cdot 10^{-5}$	$1.9 \cdot 10^{-5}$	$7.7 \cdot 10^{-5}$	$2.0 \cdot 10^{-4}$	$1.6 \cdot 10^{-4}$
$\text{mean}_{\boldsymbol{\mu}}\ \epsilon_{rel}(\boldsymbol{\mu})$	$9.7 \cdot 10^{-3}$	$5.0 \cdot 10^{-3}$	$8.3 \cdot 10^{-3}$	$1.3 \cdot 10^{-2}$	$1.1 \cdot 10^{-2}$

Table 4.3: Test case 1. Computational data related to DEIM-based and DNN-based hyper-ROMs, for $N = 4$.

In terms of efficiency, the DNN-based ROMs outperform the POD-Galerkin-DEIM ROMs substantially. In particular, both Newton and Broyden approaches are able to compute the reduced solutions in less than 0.03 s, thus yielding an overall speed-up of order $\mathcal{O}(10^3)$ compared to the FOM. For what concerns the approximation error, Deep-HyROMnet shows better results with respect to both deep learning-based Broyden approaches, and is able to achieve the same accuracy of DEIM-22, whilst being almost 100 times faster. The evolution of the $L^2(\Omega_0)$-absolute error, averaged over the testing parameters, is reported in Figure 4.3 for all of the hyper-ROMs considered.

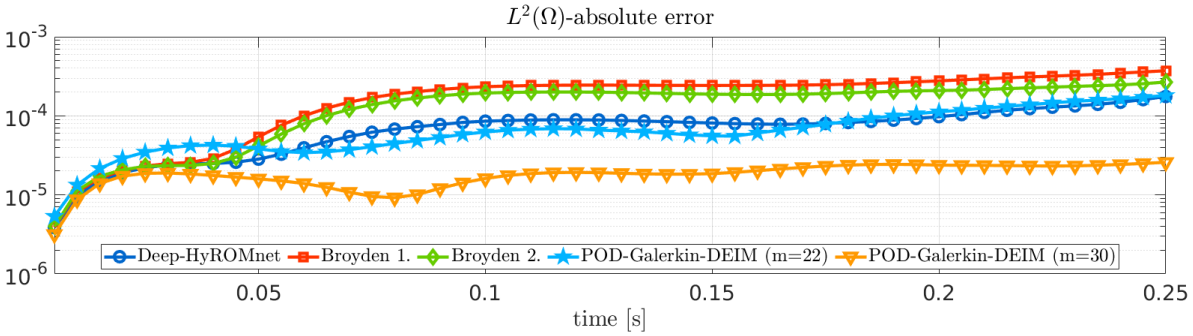
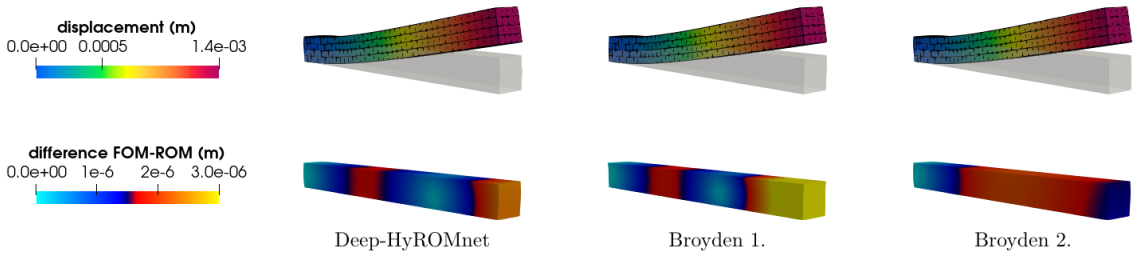


Figure 4.3: Test case 1. Evolution in time of the average $L^2(\Omega_0)$-absolute error computed using DEIM-based and DNN-based hyper-ROMs, for $N = 4$.

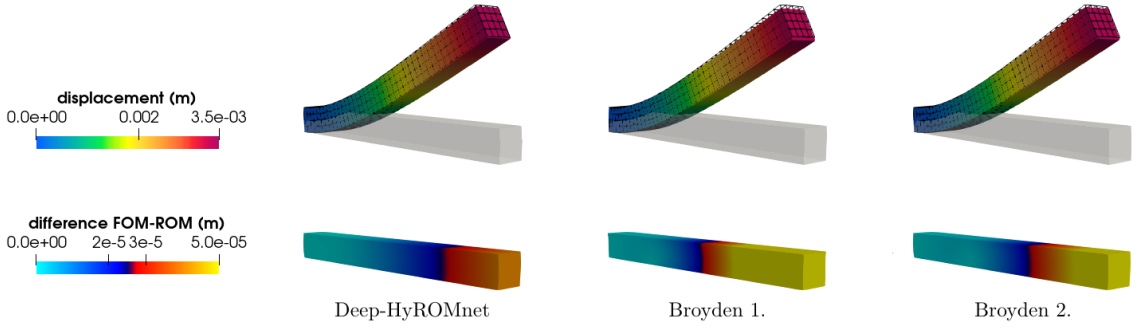
	Deep-HyROMnet	Broyden 1.	Broyden 2.
$N = 4$	$7.7 \cdot 10^{-5}$	$2.0 \cdot 10^{-4}$	$1.6 \cdot 10^{-4}$
$N = 8$	$9.0 \cdot 10^{-5}$	$2.1 \cdot 10^{-4}$	$1.8 \cdot 10^{-4}$
$N = 12$	$1.4 \cdot 10^{-4}$	$2.7 \cdot 10^{-4}$	$2.1 \cdot 10^{-4}$
$N = 16$	$1.8 \cdot 10^{-4}$	$3.3 \cdot 10^{-4}$	$2.9 \cdot 10^{-4}$

Table 4.4: Test case 1. Absolute error of DNN-based hyper-ROMs, for different RB dimensions N.

The final accuracy of the hyper-ROMs equals that of the ROM without hyper-reduction, i.e. $\epsilon_{rel} \approx 10^{-2}$ (see Figure 2.4), meaning that the projection error dominates over the nonlinear operators approximation error. In order to increase the accuracy of the reduced solution, we should consider higher values of the RB dimension N. However, by increasing the RB dimension, the task of the DNNs becomes more complex, meaning that more training samples and a larger size of the neural networks themselves may be required. In Table 4.4 we report the average absolute error ϵ_{abs} associated with the hyper-ROMs when N is increased, but the number of training snapshots and the architecture of the DNNs are unchanged. Figures 4.4a and 4.4b report the FOM and the DNN-based hyper-ROMs displacements at time $T = 0.25$ s in two scenarios, together with the error between the high-fidelity solution and the reduced solution.



(a) $\boldsymbol{\mu} = [1.3225 \cdot 10^4 \text{ Pa}, 3.9875 \cdot 10^4 \text{ Pa}, 3.43 \text{ Pa}]$.



(b) $\boldsymbol{\mu} = [0.6625 \cdot 10^4 \text{ Pa}, 5.8625 \cdot 10^4 \text{ Pa}, 4.89 \text{ Pa}]$.

Figure 4.4: Test case 1. FOM (wireframe) and DNN-based hyper-ROMs (colored) displacements and corresponding difference at time $T = 0.25$ s.

Test case 2: hat function for the pressure load

Let us now consider the case of a piecewise linear pressure load describing a structure which is increasingly loaded until a maximum pressure is reached and then linearly unloaded in order to recover the initial resting state, so that

$$\mathbf{g}(t;\boldsymbol{\mu}) = \widetilde{p}\ \left(2t\ \chi(t)_{\left(0,\frac{T}{2}\right]} + 2(T-t)\ \chi(t)_{\left(\frac{T}{2},T\right]}\right), \quad t \in (0,T).$$

We recall that $T = 0.35$ s and $\Delta t = 5 \cdot 10^{-3}$ s, whereas we consider the maximum external load $\widetilde{p} \in [2, 12]$ Pa as varying parameter. The shear modulus G and the bulk modulus K are fixed to the values 10^4 Pa and $5 \cdot 10^4$ Pa, respectively. The RB basis $\mathbf{V}$ is built by performing POD on $n_s = 50$ FOM solution snapshots with tolerance $\varepsilon_{POD} = 10^{-4}$, thus obtaining $N = 4$, whilst the training set for the DNNs is constructed from $n_s' = 300$ POD-Galerkin ROM simulations.

In order to compare the computational speed-up and the approximation accuracy of the deep learning-based ROMs with the ones of POD-Galerkin-DEIM ROM, we choose DEIM-14 and DEIM-29 (that are, the DEIM-based hyper-ROMs associated with $m = 14$ and $m = 29$, respectively), since these dimensions leads to POD-Galerkin-DEIM ROMs with a good trade-off between accurate results and required CPU time. Table 4.5 shows the comparison between the

	DEIM-22	DEIM-30	Deep-HyROMnet	Broyden 1.	Broyden 2.
Speed-up	$\times 14$	$\times 9$	$\times 1133$	$\times 1126$	$\times 1153$
Avg. CPU time	3 s	5 s	0.035 s	0.036 s	0.036 s
$\text{mean}_{\boldsymbol{\mu}}\ \epsilon_{abs}(\boldsymbol{\mu})$	$9.0 \cdot 10^{-5}$	$6.8 \cdot 10^{-6}$	$2.0 \cdot 10^{-4}$	$1.9 \cdot 10^{-3}$	$1.9 \cdot 10^{-3}$
$\text{mean}_{\boldsymbol{\mu}}\ \epsilon_{rel}(\boldsymbol{\mu})$	$1.5 \cdot 10^{-2}$	$1.4 \cdot 10^{-3}$	$1.7 \cdot 10^{-2}$	$1.6 \cdot 10^{-1}$	$1.7 \cdot 10^{-1}$

Table 4.5: Test case 2. Computational data related to DEIM-based and DNN-based hyper-ROMs, for $N = 4$.

hyper-reduced models on a testing set of 50 parameter instances. As observed in the previous test case, Deep-HyROMnet is able to achieve good results in terms of accuracy, comparable with the fastest DEIM-based model, at a greatly reduced cost. Also in this case, the speed-up achieved by the DNN-based hyper-ROMs is of order $\mathcal{O}(10^3)$ with respect to the FOM, since less than 0.04 s are needed to compute the reduced solution for each new instance of the parameter, against a time of about 40 s required by the FOM, and of 3 s required by DEIM-14. The evolution in time of the average $L^2(\Omega_0)$-absolute error is shown in Figure 4.5 for all of the hyper-ROMs considered.

We point out that the accuracy obtained using Deep-HyROMnet, although slightly lower than the ones achieved using a DEIM-based approximation, is still satisfying in all the considered scenarios. On the other hand, for some parameter values, the Broyden-based models do not lead to accurate approximations, as shown in Figures 4.6 and 4.7, where the FOM and the DNN-based hyper-ROMs displacements at time $t = 0.19$ s (i.e. corresponding to the greatest deformation) for two different values of the pressure parameter $\widetilde{p}$ are reported, together with the error between the high-fidelity and the reduced solutions. Accordingly to these results, from now on we will consider only the Deep-HyROMnet strategy among the DNN-based hyper-ROMs.

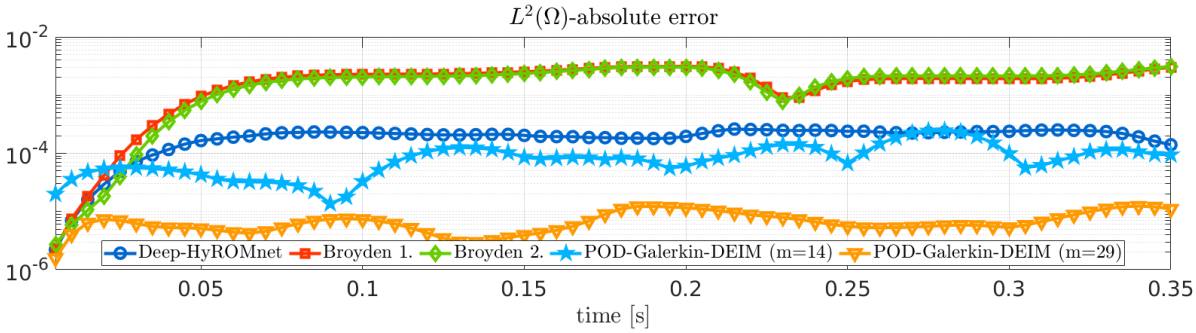


Figure 4.5: Test case 2. Evolution in time of the average $L^2(\Omega_0)$-absolute error computed using DEIM-based and DNN-based hyper-ROMs, for $N = 4$.

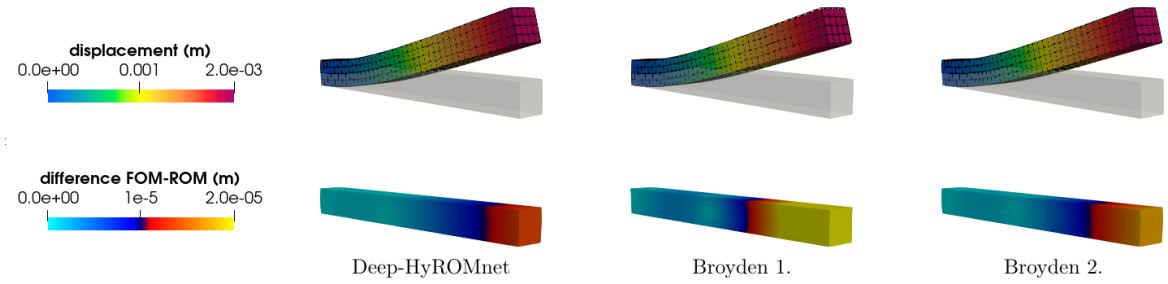


Figure 4.6: Test case 2. FOM (wireframe) and DNN-based hyper-ROMs (colored) displacements and corresponding difference at time $t = 0.19$ s for $\boldsymbol{\mu} = [3.7625$ Pa$]$.

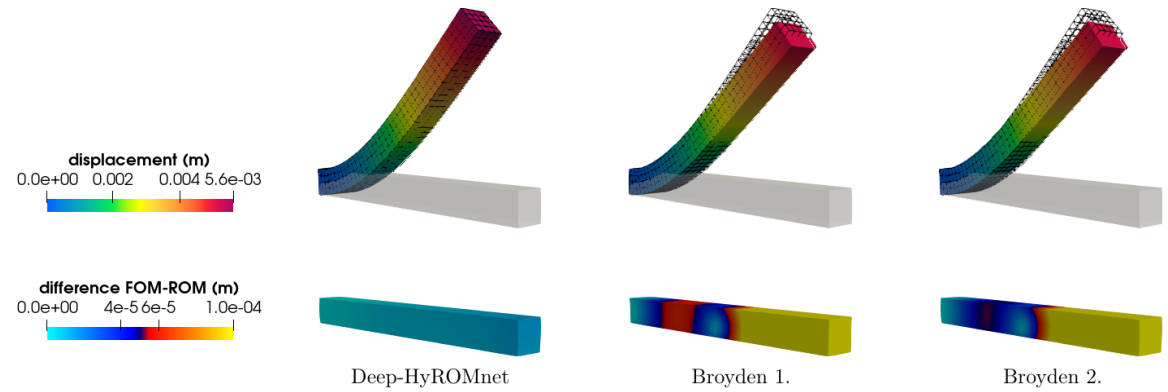


Figure 4.7: Test case 2. FOM (wireframe) and DNN-based hyper-ROMs (colored) displacements and corresponding difference at time $t = 0.19$ s for $\boldsymbol{\mu} = [10.7375$ Pa$]$.

Test case 3: step function for the pressure load

As last test case for the clamped beam, we consider a constant pressure load acting on the bottom surface area for $t \in (0, T/3]$, being $T = 0.27$ s the final simulation time, i.e.

$$\mathbf{g}(t; \boldsymbol{\mu}) = \widetilde{p}\ \chi(t)_{(0, \frac{T}{3}]},$$

so that the deformation of the object results in several oscillations of decreasing amplitude. Let $\Delta t = 3.6 \cdot 10^{-3}$ s be a uniform time step and the parameter $\boldsymbol{\mu}$ be equal to the external load $\widetilde{p} \in [2, 12]$ Pa, whilst $G = 10^4$ Pa and $K = 5 \cdot 10^4$ Pa are fixed. We build the reduced basis $\mathbf{V} \in \mathbb{R}^{N_h \times N}$ from a training set of $n_s = 50$ FOM solutions using $\varepsilon_{POD} = 10^{-3}$, thus obtaining a reduced dimension of $N = 4$, and perform POD-Galerkin ROM simulations for a given set of $n_s' = 300$ parameter samples to collect the nonlinear terms data necessary for the construction of both POD-Galerkin-DEIM and Deep-HyROMnet ROMs.

Like for the previous test cases, we compare POD-Galerkin-DEIM and Deep-HyROMnet ROMs, with respect to the displacement error and the CPU time ratio. As reported in Table 4.6, also in this case Deep-HyROMnet outperforms DEIM substantially in terms of efficiency when handling the nonlinear terms. Indeed, Deep-HyROMnet yields a ROM that is more than 1000 times faster than the FOM (this latter requiring 51 s in average to be solved), still providing satisfactory results in terms of accuracy. Figures 4.8 and 4.9 represent the Deep-HyROMnet solution at different time instants for two different values of the parameter and show that the hyper-ROM is able to correctly capture the nonlinear behavior of the continuum body also when the inertial term cannot be neglected.

	DEIM ($m = 18$)	DEIM ($m = 38$)	Deep-HyROMnet
Computational speed-up	$\times 12$	$\times 6$	$\times 1350$
Avg. CPU time	4 s	8 s	0.038 s
Time-avg. $L^2(\Omega_0)$-absolute error	$2.4 \cdot 10^{-3}$	$1.3 \cdot 10^{-4}$	$4.8 \cdot 10^{-4}$
Time-avg. $L^2(\Omega_0)$-relative error	$6.7 \cdot 10^{-1}$	$2.4 \cdot 10^{-2}$	$1.0 \cdot 10^{-1}$

Table 4.6: Test case 3. Computational data related to POD-Galerkin-DEIM ROMs and Deep-HyROMnet, for $N = 4$.

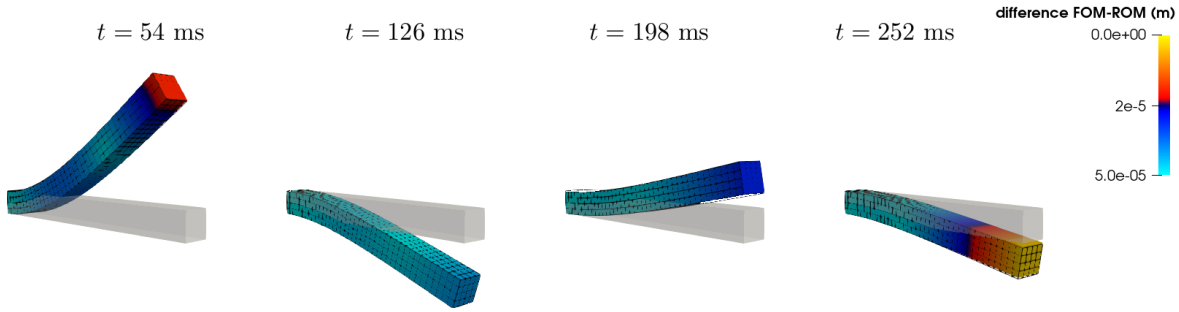


Figure 4.8: Test case 3. FOM (wireframe) and Deep-HyROMnet (colored) solutions computed at different times for $\boldsymbol{\mu} = [5.1125$ Pa$]$.

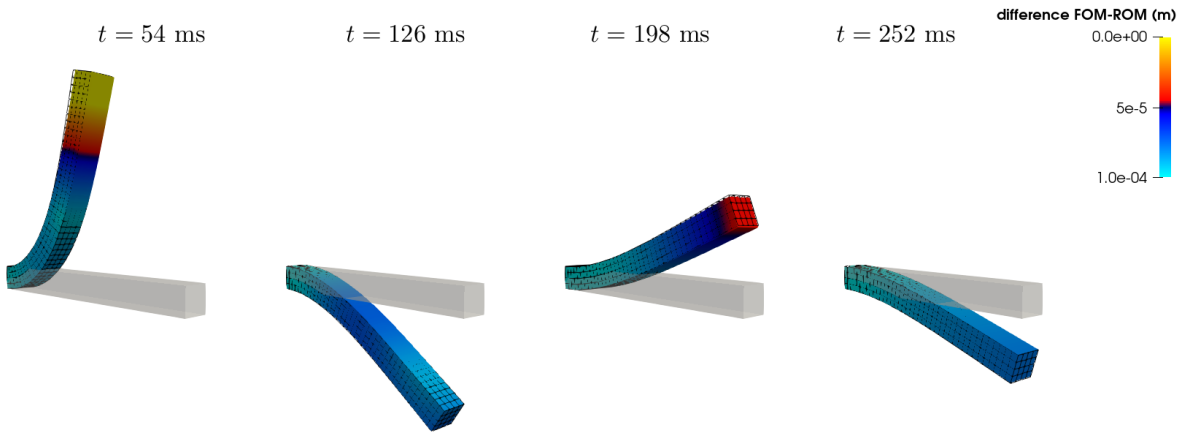


Figure 4.9: Test case 3. FOM (wireframe) and Deep-HyROMnet (colored) solutions computed at different times for $\boldsymbol{\mu} = [11.4625 \text{ Pa}]$.

4.2.2 Benchmark problems with a prolate spheroid geometry

In order to investigate the performances of Deep-HyROMnet for the solution to cardiac mechanics problems, we consider the benchmark test cases introduced in Section 2.3.2, where the reference geometry $\Omega_0 \subset \mathbb{R}^3$ is given by an idealized left ventricle (see Figure 4.10, here reported for the sake of convenience) and the nearly-incompressible Guccione relation (1.3) is adopted as constitutive equation.

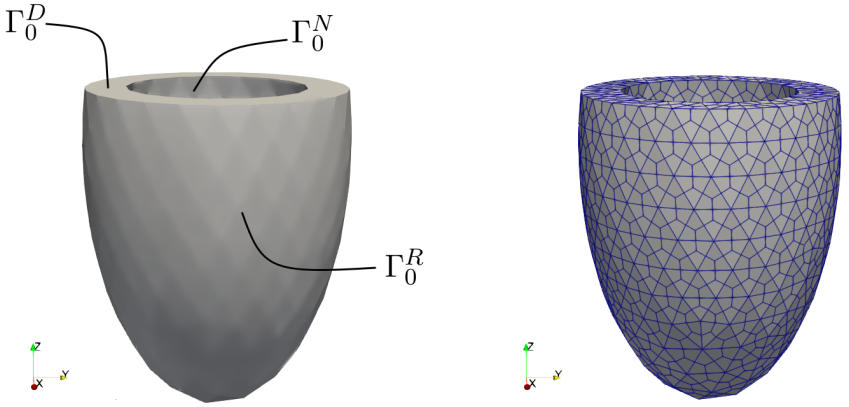


Figure 4.10: Idealized truncated ellipsoid geometry (left) and computational grid (right).

A linear external pressure

$$\mathbf{g}(t; \boldsymbol{\mu}) = \widetilde{p}\, t/T, \quad t \in (0, T),$$

with $\widetilde{p} > 0$, is applied at the endocardium (i.e., Γ_0^N) to simulate the presence of blood inside the ventricle, while the base (i.e., Γ_0^D) is kept fixed through homogeneous Dirichlet boundary conditions. The FOM is built on a hexahedral mesh with 4804 elements and 6455 vertices, so that the high-fidelity dimension is $N_h = 19365$ when $\mathbb{Q}_1$-FE are employed.

Passive inflation of a left ventricle

In order to simulate the passive ventricular diastole we do not account for anisotropy and active tension, and only consider the passive component of the first Piola-Kirchhoff stress tensor. As unknown parameters, we choose

- the material stiffness in fiber and cross-fiber directions $b_f, b_n \in [0.8, 1.2]$, and
- the multiplicative factor $C \in [8 \cdot 10^3, 12 \cdot 10^3]$ Pa,

whilst all other parameters are fixed to their reference values. For the time setting, we assume $t \in (0, 0.25)$ s and a uniform time step $\Delta t = 5 \cdot 10^{-3}$ s. We recall that almost 6 minutes are required by the FOM to compute the solution for each new instance of the input parameter.

In order to compute the RB basis $\mathbf{V} \in \mathbb{R}^{N_h \times N}$, we collect the solution snapshots for $n_s = 50$ points sampled in the parameter space $\mathcal{P}$ and apply POD with $\varepsilon_{POD} = 10^{-5}$, obtaining a reduced basis of dimension $N = 16$. Furthermore, with the aim of training the DNNs, we collect data from $n_s' = 50$ POD-Galerkin ROM simulations. For comparison purposes, we consider POD-Galerkin-DEIM hyper-ROMs built with $m = 84$ and $m = 127$, in order to balance accuracy and efficiency. The evolution in time of the average $L^2(\Omega_0)$-absolute error and the computational data obtained over a testing set of 20 parameters are reported in Figure 4.11 and Table 4.7, respectively, for all of the hyper-ROMs considered.

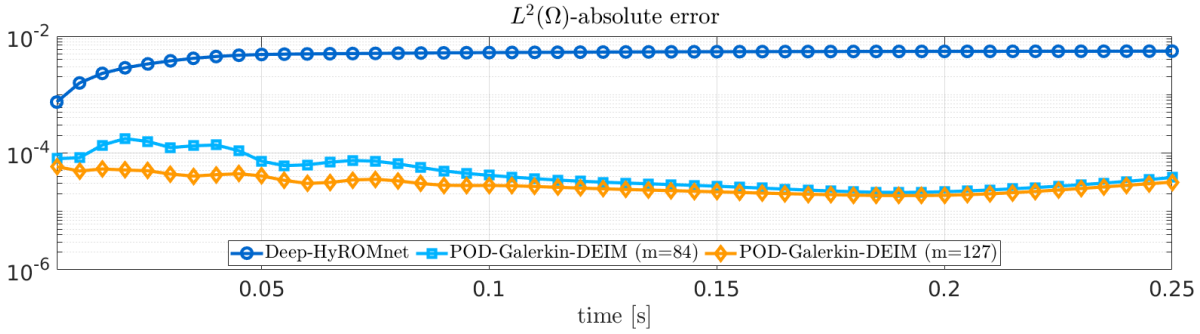


Figure 4.11: Passive inflation of a left ventricle. Evolution in time of the average $L^2(\Omega_0)$-absolute error computed using DEIM-based and DNN-based hyper-ROMs, for $N = 16$.

	DEIM ($m = 84$)	DEIM ($m = 127$)	Deep-HyROMnet
Computational speed-up	$\times 9$	$\times 8$	$\times 3360$
Avg. CPU time	36 s	42 s	0.1 s
Time-avg. $L^2(\Omega_0)$-absolute error	$5.3 \cdot 10^{-5}$	$2.9 \cdot 10^{-5}$	$4.8 \cdot 10^{-3}$
Time-avg. $L^2(\Omega_0)$-relative error	$5.3 \cdot 10^{-4}$	$2.9 \cdot 10^{-4}$	$1.9 \cdot 10^{-2}$

Table 4.7: Passive inflation of a left ventricle. Computational data related to POD-Galerkin-DEIM ROMs and Deep-HyROMnet, for $N = 16$.

In this scenario Deep-HyROMnet is able to achieve a speed-up of 3360 times with respect to the FOM, allowing us to compute the reduced solution in 0.1 s, whereas at least 36 s are required by the DEIM-based hyper-ROM. We recall, in fact, that no further speed-ups can be achieved by POD-Galerkin-DEIM by decreasing m, thus the size of the reduced mesh. The DNN-based computed solutions, despite being less accurate than those computed by means of classical hyper-reduction techniques, show good approximation performances. Figure 4.12 shows the Deep-HyROMnet solution at time $T = 0.25$ s, for three different values of the parameter, together with the corresponding pointwise error with respect to the FOM.

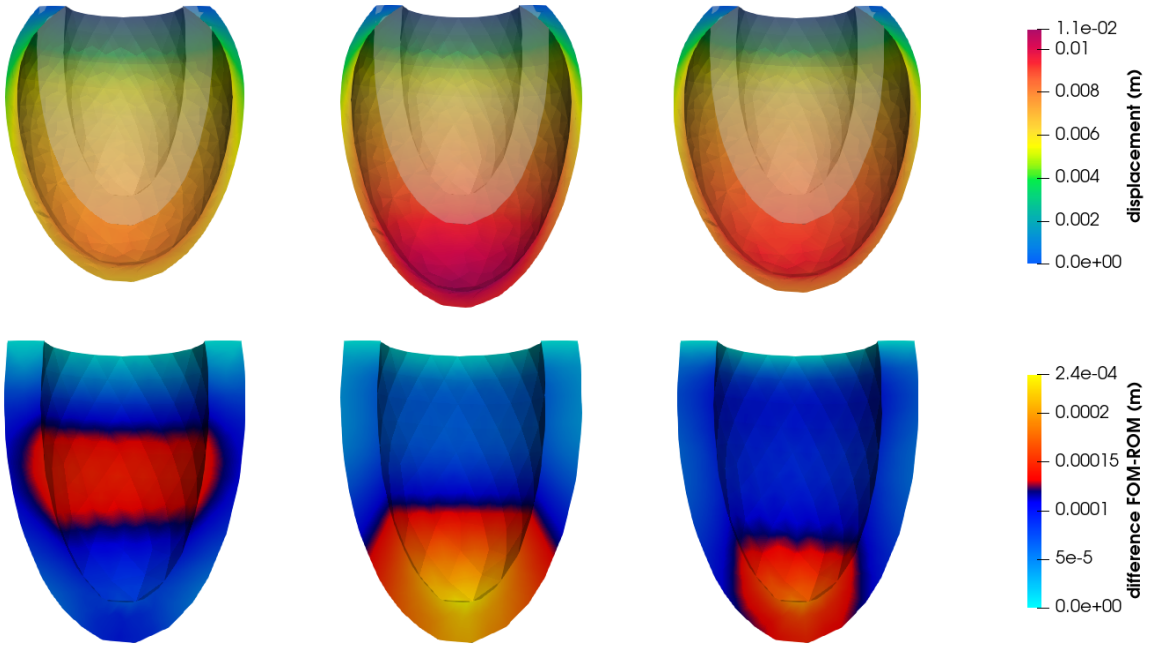


Figure 4.12: Passive inflation of a left ventricle. Deep-HyROMnet displacements (top) and pointwise error (bottom) for $\boldsymbol{\mu} = [0.881, 1.171, 11.65 \cdot 10^3 \text{ Pa}]$ (left), $\boldsymbol{\mu} = [1.007, 0.853, 8.61 \cdot 10^3 \text{ Pa}]$ (center) and $\boldsymbol{\mu} = [0.827, 1.049, 8.15 \cdot 10^3 \text{ Pa}]$ (right).

Passive inflation and active contraction of a left ventricle

Finally, we assess the performances of the reduced model for the simulation of ventricular systole. With respect to the previous test case, we take into account a varying fiber distribution and contractile forces by introducing a parameterized active tension of the form

$$T_a(t; \boldsymbol{\mu}) = \widetilde{T}_a \; t/T, \quad t \in (0, T),$$

in the material law. We recall that, as varying parameters, we consider

- the maximum value of the active tension $\widetilde{T}_a \in [49.5 \cdot 10^3, 70.5 \cdot 10^3]$ Pa, and
- the fiber angles $\boldsymbol{\alpha}^{epi} \in [-105.5, -74.5]^\circ$ and $\boldsymbol{\alpha}^{endo} \in [74.5, 105.5]^\circ$.

All other model inputs are fixed to their reference values, namely $b_f = 8$, $b_s = b_n = b_{sn} = 2$, $b_{fs} = b_{fn} = 4$, $C = 2 \cdot 10^3$ Pa, $K = 50 \cdot 10^3$ Pa and $\widetilde{p} = 15 \cdot 10^3$. Regarding the time discretization,

we choose $t \in (0, 0.25)$ s and a uniform $\Delta t = 5 \cdot 10^{-3}$ s, resulting in a total number of 50 time steps. The FOM is built on a hexahedral mesh of a truncated ellipsoid with 4804 elements and 6455 vertices, so that $N_h = 19365$ when $\mathbb{Q}_1$-FE are employed. In this case almost 6 min are required to compute a single high-fidelity solution. The reduced basis $\mathbf{V} \in \mathbb{R}^{N_h \times N}$ is computed by means of the POD method on a set of FOM solutions computed for 50 points sampled in the parameter space $\mathcal{P}$, resulting in a RB dimension equals to $N = 16$ when choosing $\varepsilon_{POD} = 10^{-3}$. Moreover, we run $n_s = 200$ simulations of the ROM without hyper-reduction in order to collect the necessary training sets for the DNNs, as well as the residual snapshots required for the construction of the DEIM-based hyper-ROM.

Table 4.8 reports the computational data of the POD-Galerkin-DEIM ROM, in the most efficient configurations, and of the Deep-HyROMnet, clearly showing that the latter outperforms the classical reduction strategy regarding the computational speed-up. In fact, Deep-HyROMnet is able to approximate the solution dynamics in 0.1 s, that is even faster than real-time, while a POD-Galerkin-DEIM ROM requires 1 min in average, where the final simulation time T is set equal to 0.25 s. Although the Deep-HyROMnet error is one order of magnitude higher than the one evaluated by a DEIM-based hyper-ROM (see Figure 4.13), the results are satisfactory in terms of accuracy. In Figures 4.14 the FOM and the DNN-based hyper-ROM displacements at time $T = 0.25$ s are reported for three different values of the parameters, together with the error between the high-fidelity and the reduced solutions.

	DEIM-303	DEIM-456	Deep-HyROMnet
Speed-up	$\times 6$	$\times 5$	$\times 3554$
Avg. CPU time	58 s	75 s	0.1 s
$\text{mean}_{\boldsymbol{\mu}}\ \epsilon_{abs}(\boldsymbol{\mu})$	$1.3 \cdot 10^{-3}$	$6.6 \cdot 10^{-4}$	$1.5 \cdot 10^{-2}$
$\text{mean}_{\boldsymbol{\mu}}\ \epsilon_{rel}(\boldsymbol{\mu})$	$7.5 \cdot 10^{-3}$	$5.0 \cdot 10^{-3}$	$6.4 \cdot 10^{-2}$

Table 4.8: Passive inflation and active contraction of a left ventricle. Computational data related to DEIM-based and DNN-based hyper-ROMs, for $N = 16$.

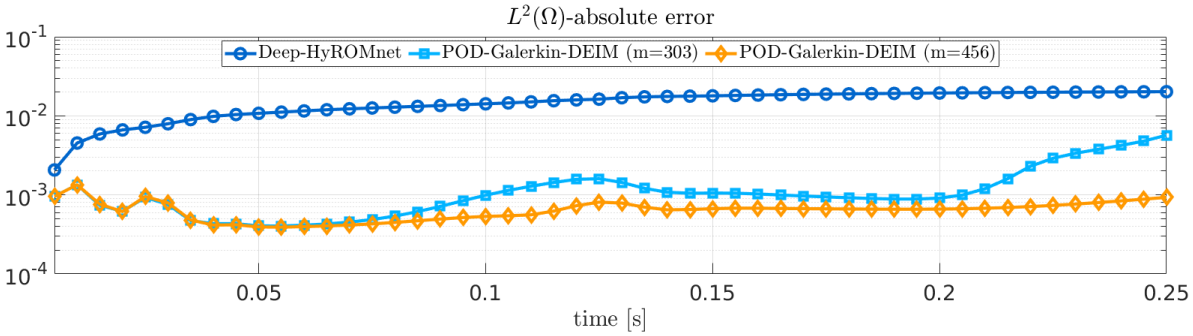


Figure 4.13: Passive inflation and active contraction of a left ventricle. Evolution in time of the average $L^2(\Omega_0)$-absolute error computed using DEIM-based and DNN-based hyper-ROMs, for $N = 16$.

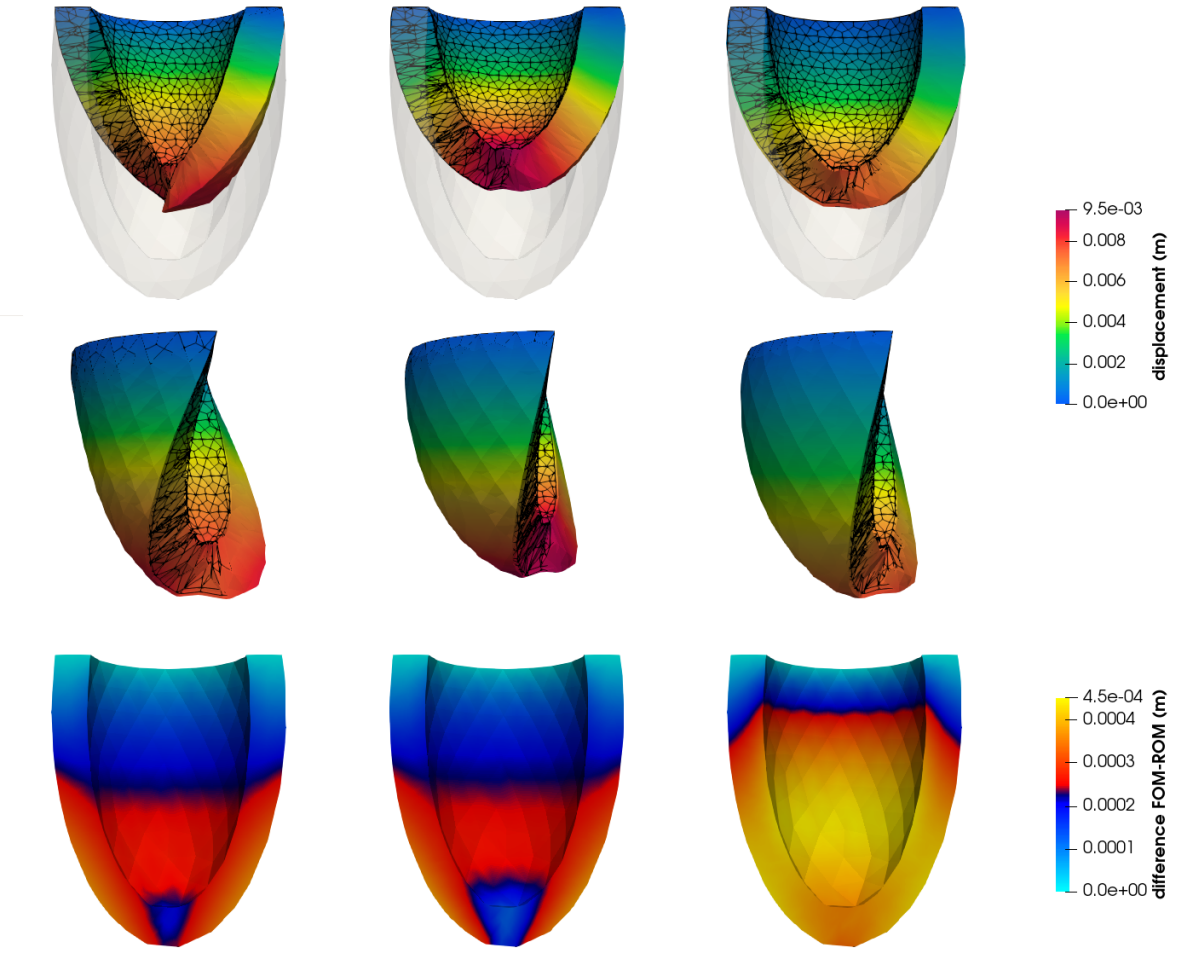


Figure 4.14: Passive inflation and active contraction of a left ventricle. FOM (wireframe) and Deep-HyROMnet (colored) displacements (frontal view on top, lateral view in the middle) and corresponding difference (bottom) at time $T = 0.25$ s for $\boldsymbol{\mu} = [61942.5 \text{ Pa}, -77.5225°, 87.9075°]$ (left), $\boldsymbol{\mu} = [59737.5 \text{ Pa}, -102.3225°, 91.1625°]$ (center) and $\boldsymbol{\mu} = [50497.5 \text{ Pa}, -100.9275°, 80.0025°]$ (right).

To conclude, we repeat that the approximation of the reduced nonlinear operators with Deep-HyROMnet does not depend directly on the high-fidelity dimension N_h, but rather on reduced basis dimension N. To test its performances using a higher FOM dimension, we address the solution to the problem described in this section, however considering a finer hexahedral mesh with 9964 elements and 13025 vertices, thus obtaining $N_h = 39075$ as FOM dimension. In this case about 13 minutes are required to compute the high-fidelity dynamics. On the other hand, a reduced basis of dimension $N = 16$ is computed for $\varepsilon_{POD} = 10^{-3}$. The computational data, averaged over a testing set of 20 parameter samples, are reported in Table 4.9. Almost unexpectedly, the online CPU time required by Deep-HyROMnet doubles as we double N_h. This may be due to the higher time required to perform matrix-vector multiplication for the reconstruction of the reduced solutions $\mathbf{V}\mathbf{u}_N^n(\boldsymbol{\mu})$, for $n = 1, \dots, N_t$. Further analysis should be performed to investigate this issue. Nonetheless, is it worth saying that the overall computational

speed-up of Deep-HyROMnet increases as the FOM dimension N_h grows, while the number N of reduced basis function remains small, so that reduced solutions can be computed extremely fast. For what concerns the approximation accuracy of the hyper-ROMs with respect to the associate FOMs, we obtain almost the same results, showing that Deep-HyROMnet is able to deal with higher high-fidelity dimensions.

	Deep-HyROMnet	
N_h	19365	39075
FOM time	5 min 54 s	13 min 01 s
N	16	
Speed-up	$\times 3554$	$\times 3886$
Avg. CPU time	0.1 s	0.2 s
$\text{mean}_{\boldsymbol{\mu}}\ \epsilon_{abs}(\boldsymbol{\mu})$	$1.5 \cdot 10^{-2}$	$2.7 \cdot 10^{-2}$
$\text{mean}_{\boldsymbol{\mu}}\ \epsilon_{rel}(\boldsymbol{\mu})$	$6.4 \cdot 10^{-2}$	$8.3 \cdot 10^{-2}$

Table 4.9: Passive inflation and active contraction of a left ventricle. Computational data related to Deep-HyROMnet for $N_h = 19365$ and $N_h = 39075$.

Chapter 5

Application to cardiac mechanics: physiological and pathological scenarios

In this chapter we address the efficient solution to parameterized cardiac mechanics problems by means of the Deep-HyROMnet hyper-reduced model developed in Chapter 4. In particular, we introduce a 3D-0D coupled structure-circulation model for the left ventricle in Section 5.1, taking into account contractile forces by means of a surrogate active tension. Further, we show how our DNN-based hyper-ROM can be adapted to different test cases in this context in Section 5.2. To conclude, we present two numerical examples in Section 5.3 on a patient-specific left ventricle geometry, first focusing on a physiological scenario (Section 5.3.1) and then assuming the presence of an ischemic region inside the myocardium (Section 5.3.2), also showing, in Section 5.3.3, how the developed framework is able to enhance forward UQ analysis otherwise unaffordable.

5.1 Mathematical models: 3D-0D coupled cardiovascular model

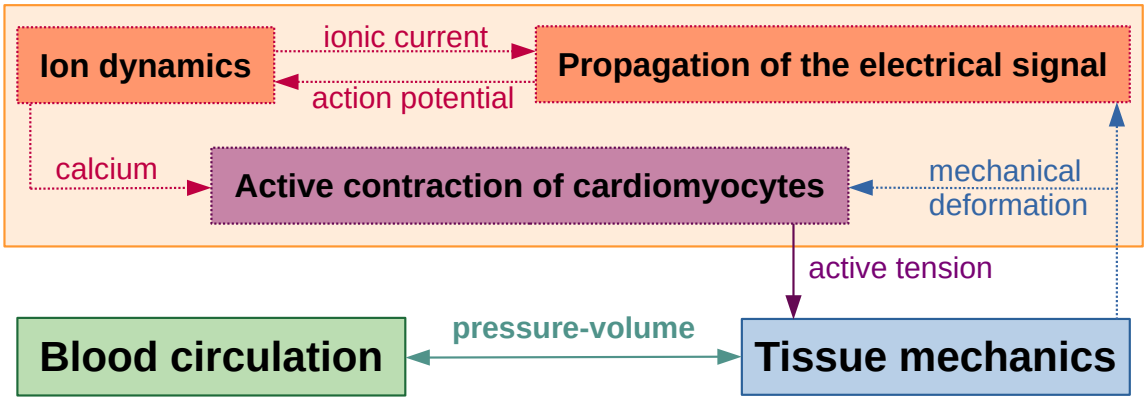


Figure 5.1: Cardiac core models describing the different physics.

When interested in the accurate and efficient solution to cardiac mechanics problems, described in Chapter 1, one needs to take into account the interactions between several biophysical phenomena concurring to the heart function, namely electrophysiology, biochemistry, mechanics and fluid dynamics, each described by suitable models (*core models*, see Figure 5.1) written in terms of PDEs and/or ODEs [QMV17]. Electrophysiology corresponds to the propagation of the electrical potential and ion dynamics, and describes the electrical activity of cardiac muscle

cells; the activation of cardiomyocytes is the results of complex mechano-chemical interactions among contractile proteins [Ber01] and provides the active tension necessary to the mechanics model. In this section we introduce the 3D structural model for the cardiac tissue and the 0D model for the blood circulation required in order to reproduce the whole cardiac cycle, and further describe the construction of the analytical form of the active tension that surrogates the input provided by the active force generation model.

5.1.1 3D elastodynamics model

The strong formulation of the nonlinear parameterized IBVP for cardiac mechanics we want to solve reads as follows: given $\boldsymbol{\mu} \in \mathcal{P}$, find the displacement field $\mathbf{u}(\boldsymbol{\mu})\colon \Omega_0 \times (0,T) \to \mathbb{R}^3$ such that

$$\begin{cases} \rho_0 \ddot{\mathbf{u}}(\boldsymbol{\mu}) - \nabla_0 \cdot \mathbf{P}(\mathbf{u}(\boldsymbol{\mu}); \boldsymbol{\mu}) = \mathbf{0} & \text{in } \Omega_0 \times (0,T), \\ \mathbf{P}(\mathbf{u}(\boldsymbol{\mu}); \boldsymbol{\mu})\mathbf{N} = p_{LV}(t; \boldsymbol{\mu}) \| J\mathbf{F}^T(\mathbf{u}(\boldsymbol{\mu}); \boldsymbol{\mu})\mathbf{N} \| \mathbf{v}(t) & \text{on } \Gamma_0^{base} \times (0,T), \\ \mathbf{P}(\mathbf{u}(\boldsymbol{\mu}); \boldsymbol{\mu})\mathbf{N} = -p_{LV}(t; \boldsymbol{\mu}) J\mathbf{F}^{-T}(\mathbf{u}(\boldsymbol{\mu}))\mathbf{N} & \text{on } \Gamma_0^{endo} \times (0,T), \\ \mathbf{P}(\mathbf{u}(\boldsymbol{\mu}); \boldsymbol{\mu})\mathbf{N} + \mathbf{K}^{epi}\mathbf{u}(\boldsymbol{\mu}) + \mathbf{C}^{epi}\dot{\mathbf{u}}(\boldsymbol{\mu}) = 0 & \text{on } \Gamma_0^{epi} \times (0,T), \\ \mathbf{u}(\boldsymbol{\mu}) = \mathbf{u}_0(\boldsymbol{\mu}); \quad \dot{\mathbf{u}}(\boldsymbol{\mu}) = \dot{\mathbf{u}}_0(\boldsymbol{\mu}) & \text{in } \Omega_0 \times \{0\}, \end{cases} \tag{5.1}$$

where the computational boundary $\partial\Omega_0$ is divided into the inner endocardium Γ_0^{endo}, the outer epicardium Γ_0^{epi} and the ventricular base Γ_0^{base}, the latter representing the artificial boundary resulting from truncation of the heart below the valves in a short axis plane. The boundary conditions on Γ_0^{base} are energy-consistent and provide an explicit expression for the stresses at the base [RDQ20b], being

$$\mathbf{v}(t) = \frac{\int_{\Gamma_0^{endo}} J\mathbf{F}^{-T}(\mathbf{u}(\boldsymbol{\mu}))\mathbf{N} d\Gamma}{\int_{\Gamma_0^{base}} \| J\mathbf{F}^{-T}(\mathbf{u}(\boldsymbol{\mu}))\mathbf{N} \| d\Gamma}.$$

At the endocardium we take into account the action of the blood pressure $p_{LV} = p_{LV}(t; \boldsymbol{\mu})$ inside the chamber, which is governed by the circulation model of Section 5.1.2. Finally, the Robin boundary conditions at the epicardium aim at modeling the interaction between the ventricle and the pericardium [GDQ19], that is the fibroelastic sac containing the heart, and are given by $\mathbf{K}^{epi} = K_\perp(\mathbf{N} \otimes \mathbf{N}) + K_\parallel(\mathbf{I} - \mathbf{N} \otimes \mathbf{N})$ and $\mathbf{C}^{epi} = C_\perp(\mathbf{N} \otimes \mathbf{N}) + C_\parallel(\mathbf{I} - \mathbf{N} \otimes \mathbf{N})$, where the local values of stiffness $\mathbf{K}^{epi}$ and viscosity $\mathbf{C}^{epi}$ of the epicardial tissue, in the normal ($\perp$) and tangential ($\parallel$) directions, are reported in Table .6.

5.1.2 0D blood external circulation model

To provide meaningful numerical simulations of the left ventricle activity between two consecutive heartbeats, and then to characterize the complete cardiac cycle from a mechanical point of view, we need to suitably model the presence of blood inside the cardiac chamber providing the pressure to the endocardial boundary Γ_0^{endo} in (5.1). Several hemodynamics models have been proposed in the literature, see, e.g., [Reg+20a; BF10; WLW09; Nor+11], just to mention a few examples. Among these, in the context of coupled problems, lumped-parameter fluid models have been extensively considered [Hir+17; Pfa+20; Mol+18], since they provide good approximation results at a greatly reduced cost. In this Thesis we adopt the following lumped

description (0D model), as done in [GDQ19; RDQ20b], which is illustrated in the Wiggers diagram [Wig51] reported in Figure 5.2. Starting with systole, that is the phase in which the ventricle is full of blood and both the mitral valve and the aortic valve are closed, the four phases of the cardiac cycle are described as:

1. *isovolumetric contraction:* the endocardial pressure rapidly grows from the end-diastolic pressure (EDP) p_{ED} to the value p_{AVO} measured in the aorta, in such a way that the volume remains unchanged;

2. *ejection:* as soon as the aortic valve opens, the ejection phase starts and the evolution of the pressure $p_{LV}(t;\boldsymbol{\mu})$ is governed by a two-element windkessel model [WLW09], with capacitance C_p and resistance R_p:

$$\begin{cases} C_p\, \dot{p}_{LV}(t;\boldsymbol{\mu}) = -\dfrac{p_{LV}(t;\boldsymbol{\mu})}{R_p} - \dot{V}_{LV}(\mathbf{u}(\boldsymbol{\mu}),t;\boldsymbol{\mu}), \quad t \in (T_{AVO}, T_{AVC}], \\ p(T_{AVO};\boldsymbol{\mu}) = p_{AVO}. \end{cases} \tag{5.2}$$

   Here, T_{AVO} and T_{AVC} are the aortic valve opening and closing times, respectively, and p_{AVO} is the pressure measured in the aorta at the beginning of the ejection phase. This phase is characterized by a decrement of the volume due to the contraction of the ventricle;

3. *isovolumetric relaxation:* when the aortic valve closes, the ventricle relaxes and the pressure drops. As both the ventricular valves are closed, no change of volume is experienced;

4. *filling:* finally, as the pressure inside the ventricle falls below that in the atrium, the mitral valve opens and the ventricle begins to fill again, so that the pressure linearly increases to the EDP, concluding the cardiac cycle.

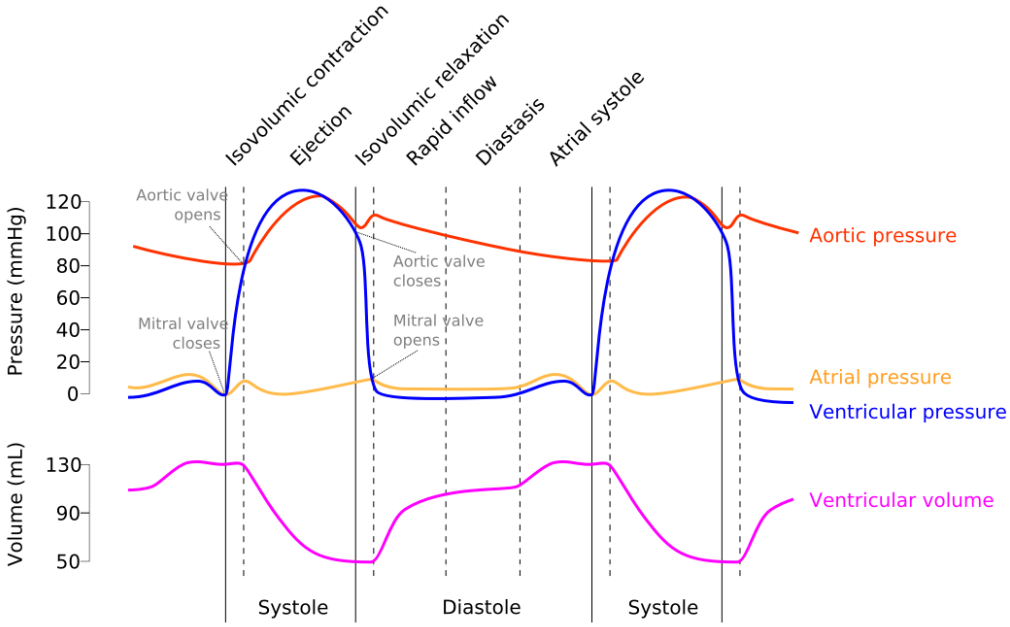


Figure 5.2: Wiggers diagram, adapted from https://commons.wikimedia.org/w/index.php?curid=50317988, illustrating the events taking place over the cardiac cycle.

5.1.3 Active tension

In order to surrogate the input provided to tissue mechanics by the active force generation model, we consider the active stress

$$\mathbf{T}_a(\mathbf{X}, t; \boldsymbol{\mu}) = T_a(t; \boldsymbol{\mu})(\mathbf{F}\mathbf{f}_0 \otimes \mathbf{f}_0) \tag{5.3}$$

acting in the fiber direction, where $T_a(t; \boldsymbol{\mu})$ is a prescribed time-dependent function, as done in test case 2.3.3 of Section 2.3. We point out that, for the sake of simplicity, we assume a uniform activation of the cardiomyocytes in the healthy tissue, while no activation of the cardiac cells will be considered inside the necrosis in pathological scenarios (see 5.3.2). For completeness, we recall the computational pipeline that is required for the definition of $T_a(t; \boldsymbol{\mu})$:

1. for a fixed set of physiological parameters, solve the 3D electromechanics problem coupled with a closed-loop circulation model by means of the FEM in space and suitable implicit-explicit schemes in time. In our case, the different core models are solved sequentially and with different time steps sizes (being the time step used for the solution to electrophysiology smaller than that used for the mechanics) [Reg+20b] in the time interval $(0, 0.8)$ s;

2. compute the space-average of the active tension coming from the EM simulation,

$$\tau_a^{EM}(t) = \underset{\mathbf{X}\in\Omega_0}{\text{avg}}\, T_a^{EM}(\mathbf{X}, t), \tag{5.4}$$

   and perform a cubic spline interpolation of $\tau_a^{EM}(t)$ to obtain the corresponding time-dependent function $\tau_a^M(t)$, reported in Figure 5.3;

3. finally, for an input parameter $\widetilde{T}_a > 0$, define

$$T_a(t; \boldsymbol{\mu}) = \frac{\widetilde{T}_a}{\max\limits_{t\in(0,0.8)s} \tau_a^M(t)}\, \tau_a^M\Big(0.8\,\Big(\frac{t}{T} - \Big\lfloor\frac{t}{T}\Big\rfloor\Big)\Big)$$

   to take into account a parameter-dependence and to model different (eventually, periodic) functions for the active tension.

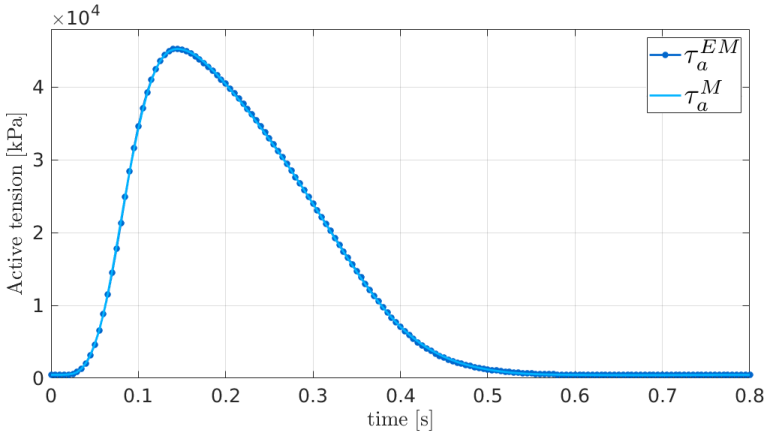


Figure 5.3: Space-averaged active tension computed during a EM simulation and the corresponding fitted curve.

To summarize, we define the active stress $T_a(t;\boldsymbol{\mu})$ in (5.3), modeling the contraction of cardiomyofiber bundles in reference fiber direction, by performing a cubic spline interpolation of the average tension computed from the solution to a 3D-0D EM problem and introducing a scaling factor depending on the parameter $\widetilde{T}_a$.

5.1.4 The 3D-0D coupled full order model

The mechanics and the blood circulation core models described in previous sections, which mutually exchange pressure and volume, must be suitable coupled in order to provide physically meaningful simulations of the cardiac cycle, leading to a 3D-0D coupled structure-windkessel problem. In this section we outline the corresponding full order model obtained by relying on the FEM in space and on implicit time schemes.

With this aim, let us denote by $\{t^0,\dots,t^{N_t}\}$ a uniform partition of the time interval $(0,T)$, corresponding to the duration of a single heartbeat, with time step Δt. To ease the notation of what follows, we define

$$(\star_h^{n,(k)}) := (\mathbf{u}_h^{n,(k)}(\boldsymbol{\mu}), p_{LV}^{n,(k)}(\boldsymbol{\mu}), t^n;\boldsymbol{\mu}),$$

where we assume $p_{LV}^{n,(k)}(\boldsymbol{\mu}) := p_{LV}^n(\boldsymbol{\mu})$, $\forall k \geq 0$, during the non-isochoric phases.

During ventricular ejection (phase 2) and filling (phase 4), the structural and the circulation problems are segregated, meaning that the two models are solved one after the other. In particular, in the ejection phase, the current pressure $p_{LV}^n(\boldsymbol{\mu}) = p_{LV}(t^n;\boldsymbol{\mu})$ is updated by solving the two-element windkessel model (5.2) before addressing the mechanics problem. For simplicity, we assume

$$\dot{V}_{LV}(\mathbf{u}_h^n(\boldsymbol{\mu}), t^n;\boldsymbol{\mu}) \approx \frac{V_{LV}(\mathbf{u}_h^{n-1}(\boldsymbol{\mu}), t^{n-1};\boldsymbol{\mu}) - V_{LV}(\mathbf{u}_h^{n-2}(\boldsymbol{\mu}), t^{n-2};\boldsymbol{\mu})}{\Delta t},$$

where the ventricular volume at time t^j, for $j \in \{n-1, n-2\}$, is computed as

$$V_{LV}(\mathbf{u}_h^j(\boldsymbol{\mu}), t^j;\boldsymbol{\mu}) = \frac{1}{3}\int_{\Gamma_0^{endo}} J(\mathbf{X} + \mathbf{u}_h^j(\boldsymbol{\mu}) - \mathbf{b}_h^j(\boldsymbol{\mu}))\cdot \mathbf{F}^{-T}\mathbf{N} d\Gamma_0,$$

being $\mathbf{b}_h^j(\boldsymbol{\mu}) = \frac{1}{|\Gamma_0^{base}|}\int_{\Gamma_0^{base}}(\mathbf{X}+\mathbf{u}_h^j(\boldsymbol{\mu}))d\Gamma_0$. For further details on the derivation of this formula we refer to [RDQ20b]. The resulting problem at time t^n, for $n = 1,\dots,N_t$, is given by the nonlinear system

$$\mathbf{R}(\mathbf{u}_h^n(\boldsymbol{\mu}), p_{LV}^n(\boldsymbol{\mu}), t^n;\boldsymbol{\mu}) = \mathbf{0} \quad \text{in } \mathbb{R}^{N_h}.$$

In order to find the unknown $\mathbf{u}_h^n(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$, we rely on the Newton method, which requires the solution of a sequence of linear systems of the form

$$\partial_{\mathbf{u}}\mathbf{R}(\star_h^{n,(k)})\delta\mathbf{u}_h^{(k)}(\boldsymbol{\mu}) = -\mathbf{R}(\star_h^{n,(k)}), \quad k \geq 0,$$

where $\partial_{\mathbf{u}}\mathbf{R}$ is the directional derivative of the structural residual. At each iteration k, the solution is updated as $\mathbf{u}_h^{n,(k+1)}(\boldsymbol{\mu}) = \mathbf{u}_h^{n,(k)}(\boldsymbol{\mu}) + \delta\mathbf{u}_h^{(k)}(\boldsymbol{\mu})$.

On the other hand, during the isovolumetric phases 1 and 3, the elastodynamics problem is solved together with the volume constraint $V_{LV}^n = V_{LV}^{n-1}$. This results in a nonlinear saddle-point system for the unknowns $\mathbf{u}_h^n(\boldsymbol{\mu})$ and $p_{LV}^n(\boldsymbol{\mu})$ of the form

$$\begin{cases} \mathbf{R}(\mathbf{u}_h^n(\boldsymbol{\mu}), p_{LV}^n(\boldsymbol{\mu}), t^n;\boldsymbol{\mu}) = \mathbf{0}, \\ V_{LV}(\mathbf{u}_h^n(\boldsymbol{\mu}), t^n;\boldsymbol{\mu}) = V_{LV}(\mathbf{u}_h^{n-1}(\boldsymbol{\mu}), t^{n-1};\boldsymbol{\mu}), \end{cases}$$

that can be solved by means of the Schur complement reduction [BGL05]. By applying Newton method, we end up with the following linear system

$$\begin{pmatrix} \partial_{\mathbf{u}}\mathbf{R}(\star_h^{n,(k)}) & \partial_p\mathbf{R}(\star_h^{n,(k)}) \\ \partial_{\mathbf{u}}\mathrm{R}^{vol}(\star_h^{n,(k)}) & 0 \end{pmatrix} \begin{pmatrix} \delta\mathbf{u}_h^{(k)}(\boldsymbol{\mu}) \\ \delta p_{LV}^{(k)}(\boldsymbol{\mu}) \end{pmatrix} = - \begin{pmatrix} \mathbf{R}(\star_h^{n,(k)}) \\ \mathrm{R}^{vol}(\star_h^{n,(k)}) \end{pmatrix}$$

at each iteration $k \geq 0$, where $\mathrm{R}^{vol} \in \mathbb{R}$ is the residual related to the volume constraint.

To summarize, the discrete nonlinear parameterized FOM for the coupled problem can be written as: given $\boldsymbol{\mu} \in \mathcal{P}$, for $n = 1, \dots, N_t$, find $\mathbf{u}_h^n(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ and $p_{LV}^n(\boldsymbol{\mu}) > 0$ such that

$$\begin{bmatrix} \mathbf{R}(\mathbf{u}_h^n(\boldsymbol{\mu}), p_{LV}^n(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}) \\ \mathrm{R}^{vol}(\mathbf{u}_h^n(\boldsymbol{\mu}), p_{LV}^n(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}) \end{bmatrix} = \mathbf{0}, \tag{5.5}$$

where R^{vol} is discarded during phases 2 and 4 of the cardiac cycle.

5.2 Deep-HyROMnet for the 3D-0D structure-circulation problem

In this section we address the reduction to the 3D-OD coupled problem described in Section 5.1.4 by means of the Deep-HyROMnet technique. We point out that, since blood circulation is modeled through a lumped-parameter model and the volume constraint implies few additional dofs to the mechanics problem, we apply model order reduction only to the structural component of (5.5), similarly to the approach adopted in [Pfa+20].

With this aim, let $\mathbf{V} \in \mathbb{R}^{N_h \times N}$ be the reduced basis built by performing POD on the snapshots matrix of mechanics displacements, i.e.

$$\mathbf{S}_u = \left[\mathbf{u}_h^1(\boldsymbol{\mu}_1) \mid \dots \mid \mathbf{u}_h^{N_t}(\boldsymbol{\mu}_1) \mid \dots \mid \mathbf{u}_h^1(\boldsymbol{\mu}_{n_s}) \mid \dots \mid \mathbf{u}_h^{N_t}(\boldsymbol{\mu}_{n_s}) \right],$$

for randomly sampled parameter values $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_{n_s}$. By performing a Galerkin projection of the FOM (5.5) onto the reduced subspace spanned by the columns of $\mathbf{V}$, we obtain the low-dimensional nonlinear problem

$$\begin{bmatrix} \mathbf{V}^T\mathbf{R}(\mathbf{V}\mathbf{u}_N^n(\boldsymbol{\mu}), p_{LV}^n(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}) \\ \mathrm{R}^{vol}(\mathbf{V}\mathbf{u}_N^n(\boldsymbol{\mu}), p_{LV}^n(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}) \end{bmatrix} = \mathbf{0}. \tag{5.6}$$

Let's define the notation

$$(\star_N^{n,(k)}) := (\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}), p_{LV}^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}).$$

The corresponding Newton system at time t^n, for $n = 1, \dots, N_t$, reads:

- for the ejection and filling phases: given an initial guess $\mathbf{u}_N^{n,(0)}(\boldsymbol{\mu})$, find $\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu})$ such that, for $k \geq 0$,

$$\begin{cases} \mathbf{V}^T\partial_{\mathbf{u}}\mathbf{R}(\star_N^{n,(k)})\mathbf{V}\delta\mathbf{u}_N^{(k)} = -\mathbf{V}^T\mathbf{R}(\star_N^{n,(k)}), \\ \mathbf{u}_N^{n,(k+1)}(\boldsymbol{\mu}) = \mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}) + \delta\mathbf{u}_N^{(k)}(\boldsymbol{\mu}), \end{cases}$$

  until $\|\mathbf{V}^T\mathbf{R}(\star_N^{n,(k+1)})\|_2 / \|\mathbf{V}^T\mathbf{R}(\star_N^{n,(0)})\|_2 < \varepsilon_{Nwt}$, where $\varepsilon_{Nwt} > 0$ is a prescribed tolerance;

- for the isovolumetric phases: given initial guesses $\mathbf{u}_N^{n,(0)}(\boldsymbol{\mu})$ and $p_{LV}^{n,(0)}(\boldsymbol{\mu})$, find $\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu})$ and $p_{LV}^{n,(k)}(\boldsymbol{\mu})$ such that, for $k \geq 0$,

$$\begin{pmatrix} \mathbf{V}^T\partial_{\mathbf{u}}\mathbf{R}(\star_N^{n,(k)})\mathbf{V} & \mathbf{V}^T\partial_p\mathbf{R}(\star_N^{n,(k)}) \\ \partial_{\mathbf{u}}\mathrm{R}^{vol}(\star_N^{n,(k)})\mathbf{V} & 0 \end{pmatrix} \begin{pmatrix} \delta\mathbf{u}_N^{(k)}(\boldsymbol{\mu}) \\ \delta p_{LV}^{(k)}(\boldsymbol{\mu}) \end{pmatrix} = - \begin{pmatrix} \mathbf{V}^T\mathbf{R}(\star_h^{n,(k)}) \\ \mathrm{R}^{vol}(\star_h^{n,(k)}) \end{pmatrix},$$

  then set $\mathbf{u}_N^{n,(k+1)}(\boldsymbol{\mu}) = \mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}) + \delta\mathbf{u}_N^{(k)}(\boldsymbol{\mu})$ and $p_{LV}^{n,(k+1)}(\boldsymbol{\mu}) = p_{LV}^{n,(k)}(\boldsymbol{\mu}) + \delta p_{LV}^{(k)}(\boldsymbol{\mu})$, until $\|\mathbf{V}^T\mathbf{R}(\star_N^{n,(k+1)})\|_2/\|\mathbf{V}^T\mathbf{R}(\star_N^{n,(0)})\|_2 < \varepsilon_{Nwt}$, where $\varepsilon_{Nwt} > 0$ is a prescribed tolerance.

We recall that $\mathbf{V}^T\mathbf{R} \in \mathbb{R}^N$, $\mathbf{V}^T\partial_{\mathbf{u}}\mathbf{R}\mathbf{V} \in \mathbb{R}^{N\times N}$, $\mathbf{V}^T\partial_p\mathbf{R} \in \mathbb{R}^N$, $\mathrm{R}^{vol} \in \mathbb{R}$ and $\partial_{\mathbf{u}}\mathrm{R}^{vol}\mathbf{V} \in \mathbb{R}^{1\times N}$, where N is the RB dimension, depend on the input parameter $\boldsymbol{\mu}$ and on the current solutions $\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu})$ and $p_{LV}^{n,(k)}(\boldsymbol{\mu})$, so that the corresponding high-fidelity arrays must be assembled at each Newton iteration before projecting them onto the reduce subspace. For this reason, we need to rely on hyper-reduction techniques.

Hence, given the triplets $\boldsymbol{\vartheta} = (\boldsymbol{\mu}, t^n, k) \in \mathcal{P} \times \{t^1, \dots, t^{N_t}\} \times \mathbb{N}^+$, we efficiently compute the N-dimensional ROM operators evaluated on $(\star_N^{n,(k)})$ by exploiting the DL-ROM-based architecture described in Chapter 4 to learn the following nonlinear maps:

$$\begin{aligned}
\boldsymbol{\rho}_N &: (\boldsymbol{\mu}, t^n, k) \longmapsto \boldsymbol{\rho}_N(\boldsymbol{\mu}, t^n, k) \approx \mathbf{V}^T\mathbf{R}(\star_N^{n,(k)}), \\
\boldsymbol{\iota}_N &: (\boldsymbol{\mu}, t^n, k) \longmapsto \boldsymbol{\iota}_N(\boldsymbol{\mu}, t^n, k) \approx \mathbf{V}^T\partial_{\mathbf{u}}\mathbf{R}(\star_N^{n,(k)})\mathbf{V}, \\
\boldsymbol{\pi}_N &: (\boldsymbol{\mu}, t^n, k) \longmapsto \boldsymbol{\pi}_N(\boldsymbol{\mu}, t^n, k) \approx \mathbf{V}^T\partial_p\mathbf{R}(\star_N^{n,(k)}), \\
\boldsymbol{\upsilon}_N &: (\boldsymbol{\mu}, t^n, k) \longmapsto [\boldsymbol{\upsilon}_N^{\partial R}(\boldsymbol{\mu}, t^n, k), \boldsymbol{\upsilon}_N^{R}(\boldsymbol{\mu}, t^n, k)] \approx [\partial_{\mathbf{u}}\mathrm{R}^{vol}(\star_N^{n,(k)})\mathbf{V}, \mathrm{R}^{vol}(\star_N^{n,(k)})].
\end{aligned}$$

The online stage of Deep-HyROMnet thus reads as follows: given $\boldsymbol{\mu} \in \mathcal{P}$, for $n = 1, \dots, N_t$, given an initial guess $\mathbf{u}_N^{n,(0)}(\boldsymbol{\mu}) = \mathbf{u}_N^{n-1}(\boldsymbol{\mu})$, for $k \geq 0$, find $\delta\mathbf{u}_N^{(k)}(\boldsymbol{\mu}) \in \mathbb{R}^N$ and $\delta p_{LV}^{(k)} > 0$ such that

$$\boldsymbol{\iota}_N(\boldsymbol{\mu}, t^n, k)\delta\mathbf{u}_N^{(k)} = -\boldsymbol{\rho}_N(\boldsymbol{\mu}, t^n, k) \tag{5.7}$$

for the ejection and filling phase, and by

$$\begin{pmatrix} \boldsymbol{\iota}_N(\boldsymbol{\mu}, t^n, k) & \boldsymbol{\pi}_N(\boldsymbol{\mu}, t^n, k) \\ \boldsymbol{\upsilon}_N^{\partial R}(\boldsymbol{\mu}, t^n, k) & 0 \end{pmatrix} \begin{pmatrix} \delta\mathbf{u}_N^{(k)} \\ \delta p_{LV}^{(k)} \end{pmatrix} = - \begin{pmatrix} \boldsymbol{\rho}_N(\boldsymbol{\mu}, t^n, k) \\ \boldsymbol{\upsilon}_N^{R}(\boldsymbol{\mu}, t^n, k) \end{pmatrix} \tag{5.8}$$

for the isovolumetric phases, until a suitable convergence criterion is fulfilled.

Thanks to the use of DNNs, the linear systems (5.7) and (5.8) are assembled in an extremely efficient way, i.e. $\mathcal{O}(10^{-3})$ s, while both FOM and POD-Galerkin ROMs require $\mathcal{O}(10^{-1})$ s or even $\mathcal{O}(1)$ s for each Newton iteration. Since this operation is done $N_t N_{nwt} \approx 800$ times during each cardiac cycle (where we consider $N_t = 320$ and an average number of Newton iterations per time step equal to $N_{nwt} = 2.5$), rely on Deep-HyROMnet allows to achieve remarkable speed-ups with respect to the FOM, about two orders of magnitude of CPU time as shown in the following section.

5.3 Numerical results

In this section we present the numerical results obtained using Deep-HyROMnet for the solution to the 3D-0D structure-windkessel model in both physiological and pathological scenarios. For the approximation of cardiac mechanics, quadratic ($\mathbb{Q}_2$) FE are commonly used [Lan+15], especially in a nearly-incompressible regime, due to possible instabilities. However, due to the huge computational costs entailed, we rely on linear ($\mathbb{Q}_1$) FE, which proved to be sufficiently accurate for the purposes at hand and less expensive, however considering suitable refined meshes. We point out that no instabilities have been observed. Moreover, we recall that the reduction strategy, acting at the algebraic level, works irrespective of the chosen FE degree. For all the numerical examples, we consider a uniform time step $\Delta t = 2.5 \cdot 10^{-3}$ s for time discretization and set the final time equal to $T = 0.8$ s, corresponding to a single heartbeat. For the construction of the fiber distribution, we employ the Bayer-Blake-Plank-Trayanova algorithm proposed in [Bay+12], depending on angles coefficients $\boldsymbol{\alpha}^{epi}$, $\boldsymbol{\alpha}^{endo}$, $\boldsymbol{\beta}^{epi}$ and $\boldsymbol{\beta}^{endo}$.

Remark 5.1
In order to correctly start the numerical simulation, we need to find the end-diastolic configuration of the left ventricle and to use the corresponding displacement as initial condition for our problem. This is done by solving the quasi-static problem (obtained from (5.1) by setting to zero the time dependent terms, see, e.g., [Reg+20a]) on the stress-free reference configuration, so that the resulting initial displacement depends on the model parameters. For practical

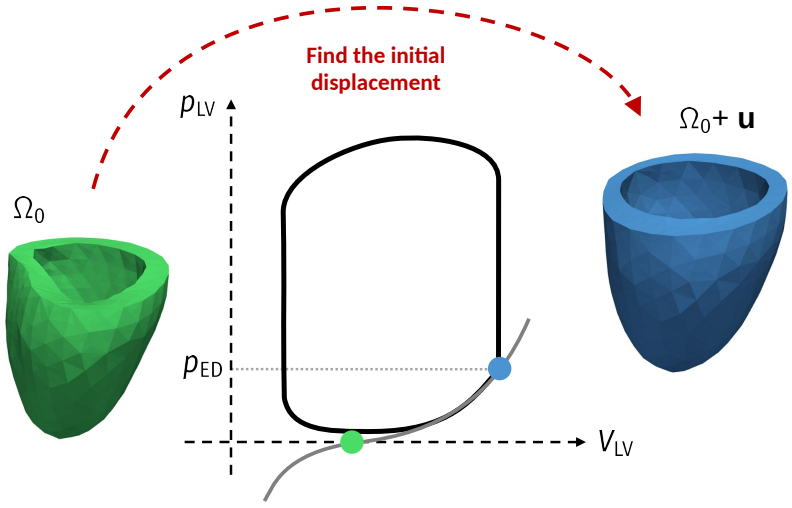


Figure 5.4: Sketch of the strategy to initialize the simulation, adapted from [Reg+20a]. The black line represents the pressure-volume loop, while the gray line is the Klotz curve [Klo+06].

reasons, we solve the initial displacement problem once and for all given the reference values of the model parameters, reported in Table .6, that we collectively denote as $\tilde{\boldsymbol{\mu}}$, so that the initial conditions are $\mathbf{u}_{h,0} = \mathbf{u}_{h,0}(\tilde{\boldsymbol{\mu}})$ and $\dot{\mathbf{u}}_{h,0} = \dot{\mathbf{u}}_{h,0}(\tilde{\boldsymbol{\mu}})$ for every instance of the parameter vector, both during training (offline stage) and testing (online stage). Nonetheless, a reduced model for the quasi-static problem can be developed in order to take into account different initial conditions as well.

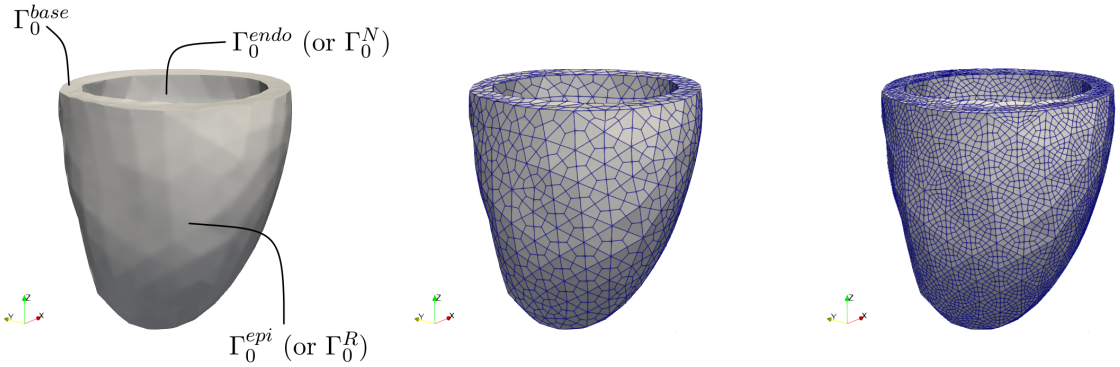


Figure 5.5: Patient-specific loaded left ventricle geometry (left) and computational grids (center and right).

In Figure 5.5 we report the computational geometry obtained when the ventricle is loaded by a value of pressure corresponding to the end diastolic pressure, in our case $p_{ED} = 15$ mmHg, and the hexahedral meshes used.

5.3.1 Physiological scenario

We now present the results of the Deep-HyROMnet approximation of the FOM solution on physiological, yet challenging, scenarios in cardiac mechanics. In this case we choose as unknown parameters

- the bulk modulus in the passive material law $K \in [2.5 \cdot 10^4, 7.5 \cdot 10^4]$ Pa,
- the resistance of the windkessel model $R_p \in [2.5 \cdot 10^7, 4.5 \cdot 10^7]$ Pa $\cdot$ s $\cdot$ m^{-3},
- the active tension parameter $\widetilde{T}_a \in [4.5 \cdot 10^4, 6 \cdot 10^4]$ Pa,

that is we set

$$\boldsymbol{\mu} = [K, R_p, \widetilde{T}_a] \in \mathcal{P} \subset \mathbb{R}^3.$$

The parameter space $\mathcal{P}$ is chosen in order to test the accuracy of Deep-HyROMnet in a wide range of scenarios. All other parameters are fixed to their reference values reported in Table .6.

The FOM is built on an hexahedral mesh with 4588 elements and 6167 vertices, featuring a high-fidelity dimension equal to $N_h = 18501$. During the offline stage we collect the solution snapshots for 20 parameter samples, each requiring almost 30 minutes using the FOM, and apply POD for the construction of the reduced basis $\mathbf{V} \in \mathbb{R}^{N_h \times N}$. In Table 5.1 we report three values of the POD tolerance ε_{POD} and the corresponding RB dimension N.

POD tolerance ε_{POD}	10^{-3}	$5 \cdot 10^{-4}$	10^{-4}
RB dimension N	39	52	99

Table 5.1: Cardiac cycle, physiological scenarios. POD tolerances and associated RB dimension for the physiological scenario, when $N_h = 18501$.

Since, as explained in Remark 4.1, the input of the encoder function is reshaped into a square matrix and we do not want to introduce too many additional terms when zero-padding, we choose N such that $\sqrt{N+1} \in \mathbb{N}$ (note that the input to $\boldsymbol{\upsilon}_N$ has dimension $N+1$), in this case $N = 63$, and build the RB basis by means of the randomized SVD, see Algorithm 3.

Once the ROM is built, we perform $n'_s = 50$ simulations to collect the reduced data necessary for training the DNNs, namely

$$\begin{aligned}
\mathbf{S}_{\boldsymbol{\rho}} &= \left[\mathbf{V}^T\mathbf{R}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}_\ell), t^n; \boldsymbol{\mu}_\ell),\ k \geq 0\right]_{n=1,\dots,N_t}^{\ell=1,\dots,n'_s}, \\
\mathbf{S}_{\boldsymbol{\iota}} &= \left[\mathbf{V}^T\partial_{\mathbf{u}}\mathbf{R}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}_\ell), t^n; \boldsymbol{\mu}_\ell)\mathbf{V},\ k \geq 0\right]_{n=1,\dots,N_t}^{\ell=1,\dots,n'_s}, \\
\mathbf{S}_{\boldsymbol{\pi}} &= \left[\mathbf{V}^T\partial_{p}\mathbf{R}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}_\ell), t^n; \boldsymbol{\mu}_\ell),\ k \geq 0\right]_{n=1,\dots,N_t}^{\ell=1,\dots,n'_s}, \\
\mathbf{S}_{\boldsymbol{\upsilon}} &= \left[\partial_{\mathbf{u}}\mathrm{R}^{vol}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}_\ell), t^n; \boldsymbol{\mu}_\ell)\mathbf{V} \mid \mathrm{R}^{vol}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}_\ell), t^n; \boldsymbol{\mu}_\ell),\ k \geq 0\right]_{n=1,\dots,N_t}^{\ell=1,\dots,n'_s},
\end{aligned} \tag{5.9}$$

where $\mathbf{S}_{\boldsymbol{\rho}} \in \mathbb{R}^{N\times 1\times N_{train}}$, $\mathbf{S}_{\boldsymbol{\iota}} \in \mathbb{R}^{N\times N\times N_{train}}$, $\mathbf{S}_{\boldsymbol{\pi}} \in \mathbb{R}^{N\times 1\times N'_{train}}$ and $\mathbf{S}_{\boldsymbol{\upsilon}} \in \mathbb{R}^{1\times(N+1)\times N'_{train}}$. Here, N_{train} and N'_{train} denote the total number of snapshots, being $N_{train} > N'_{train}$, since the snapshots for $\mathbf{S}_{\boldsymbol{\pi}}$ and $\mathbf{S}_{\boldsymbol{\upsilon}}$ are collected only during the isovolumetric phases.

A finer computational grid obtained by refining the previous mesh is also considered. In particular, 36704 elements and 42225 vertices are used, so that the average cell diameter is equal to 0.0016 m (corresponding to the mesh size commonly used to accurately capture the myocardial displacement with expensive, high-fidelity models [Aug+16; Reg+20b]). The resulting FOM is characterized by $N_h = 126675$ degrees of freedom and allows to assess the performances of Deep-HyROMnet in more realistic situations. In this case we consider $n_s = 15$ samples to build the RB basis $\mathbf{V}$ by means of randomized SVD, choosing $N = 63$ as in the coarser case. In Table 5.2 we report the RB dimension N obtained for different values of ε_{POD}. Furthermore, we perform $n'_s = 50$ ROM simulations in order to build the snapshots matrices (5.9) necessary for training the DNNs.

POD tolerance ε_{POD}	10^{-3}	$5 \cdot 10^{-4}$	10^{-4}
RB dimension N	37	48	91

Table 5.2: Cardiac cycle, physiological scenarios. POD tolerances and associated RB dimension for the physiological scenario, when $N_h = 126675$.

Remark 5.2
In order to reduce the computational time required for the training of the DNNs in the case of the finer mesh, we rely on a suitable pre-training strategy [GBC16], where the coarser model is used as initial state of the new network, as proposed in [FM22] in the case of parameterized PDE problems.

Figures 5.6, 5.7 and 5.8 show the Deep-HyROMnet solution for the coarser and the finer meshes computed at different phases of the cardiac cycle, for three different values of the parameter vector. We observe that the pointwise error between the FOM and the Deep-HyROMnet solutions does not increase in time.

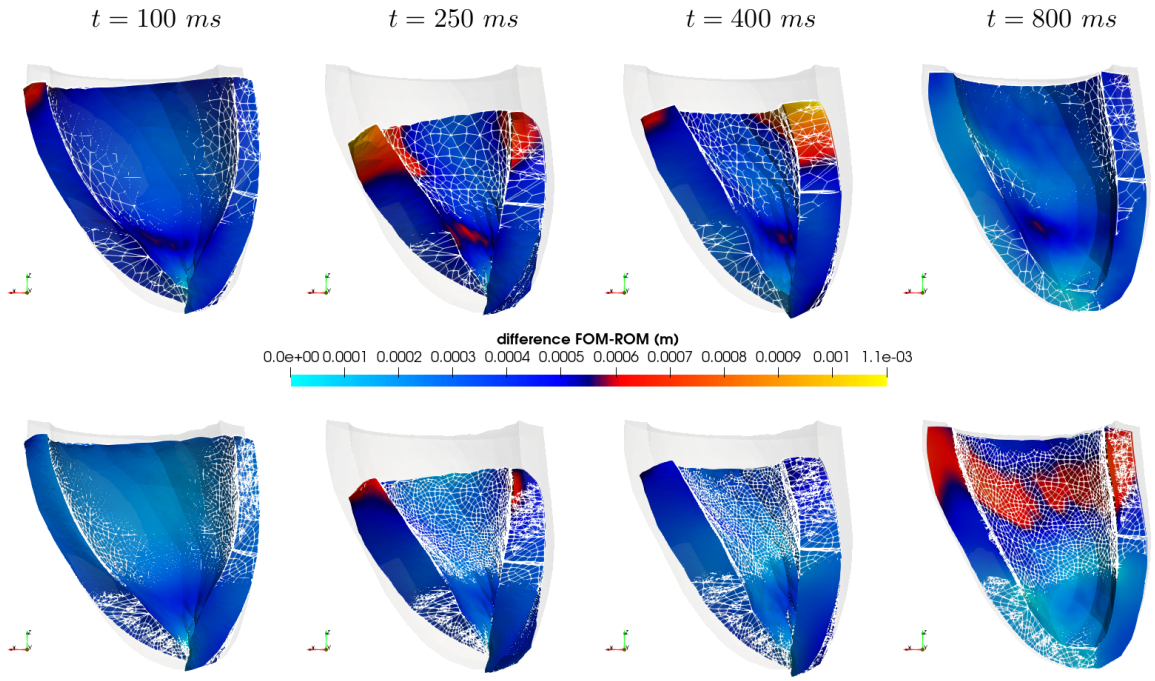


Figure 5.6: Cardiac cycle, physiological scenarios. FOM (wireframe) and Deep-HyROMnet (colored) displacements for the coarser (top) and the finer (bottom) mesh for $\boldsymbol{\mu} = [34500\ \text{Pa}, 3.00 \cdot 10^7\ \text{Pa} \cdot \text{s} \cdot \text{m}^{-3}, 55950\ \text{Pa}]$.

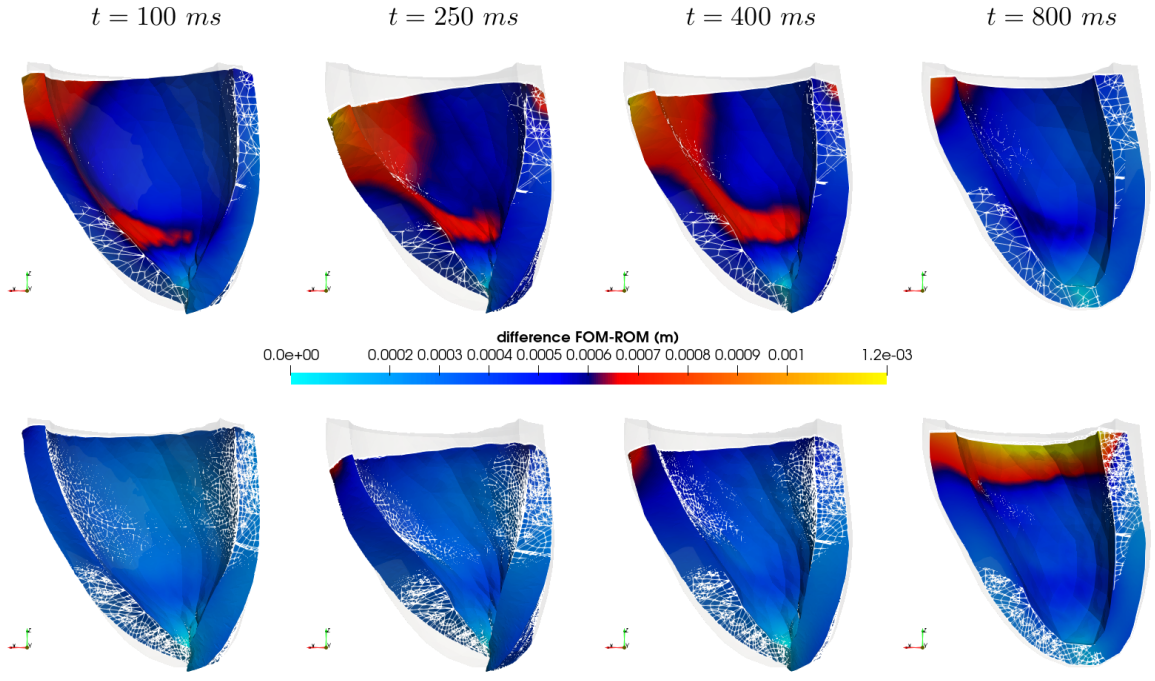


Figure 5.7: Cardiac cycle, physiological scenarios. FOM (wireframe) and Deep-HyROMnet (colored) displacements for the coarser (top) and the finer (bottom) mesh for $\boldsymbol{\mu} = [58500\ \text{Pa}, 4.16 \cdot 10^7\ \text{Pa} \cdot \text{s} \cdot \text{m}^{-3}, 49050\ \text{Pa}]$.

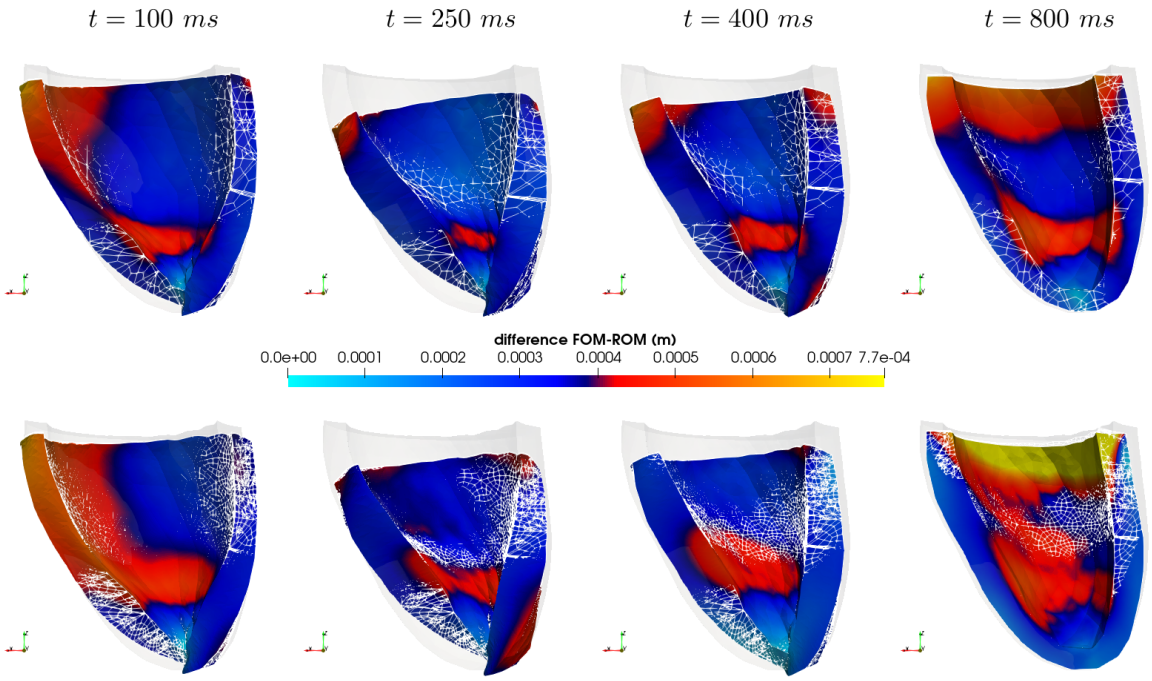


Figure 5.8: Cardiac cycle, physiological scenarios. FOM (wireframe) and Deep-HyROMnet (colored) displacements for the coarser (top) and the finer (bottom) mesh for $\boldsymbol{\mu} = [66500\ \text{Pa}, 4.20 \cdot 10^7\ \text{Pa} \cdot \text{s} \cdot \text{m}^{-3}, 57750\ \text{Pa}]$.

Table 5.3 summarizes the average results obtained on a testing set of 10 input parameters using Deep-HyROMnet for both meshes. Moreover, we report the performances of POD-Galerkin-DEIM built by employing the POD method on the ROM residual snapshots with tolerance $\varepsilon_{DEIM} = 10^{-5}$, corresponding to a DEIM residual basis of dimension $m = 1545$. No further speed-up can be achieved by decreasing the size of the reduced mesh due to convergence issues of the reduced Newton system for some instances of the considered parameters.

For $N_h = 18501$, Deep-HyROMnet computes a reduced solution in only 16 s, that is, almost

	DEIM-1045	Deep-HyROMnet	Deep-HyROMnet
N_h	18501		126675
FOM time	27 min		3 h 50 min
Speed-up	$\times 1.4$	$\times 100$	$\times 150$
Avg. CPU time	20 min	16 s	1 min 30 s
$\text{mean}_{\boldsymbol{\mu}}\ \epsilon_{abs}(\boldsymbol{\mu})$	$3 \cdot 10^{-3}$	$3 \cdot 10^{-2}$	$7 \cdot 10^{-2}$
$\text{mean}_{\boldsymbol{\mu}}\ \epsilon_{rel}(\boldsymbol{\mu})$	$7 \cdot 10^{-3}$	$7 \cdot 10^{-2}$	$7 \cdot 10^{-2}$

Table 5.3: Cardiac cycle, physiological scenarios. Computational data related to DEIM-based and DNN-based hyper-ROMs, for $N = 63$.

100 times faster than the reference high-fidelity simulation which requires 27 minutes, whilst yielding an absolute error ϵ_{abs} on the displacement field of order $O(10^{-2})$. On the other hand, the POD-Galerkin-DEIM ROM, despite being slightly more accurate than Deep-HyROMnet, still requires high computational resources, employing 20 min to simulate a single heartbeat. When using a finer mesh, Deep-HyROMnet only takes 90 s to compute the displacement dynamics for a complete heartbeat, against almost 4 h required by the FOM. We mention that the smaller speed-ups achieved with respect to the benchmark tests of Chapter 4 are due to the fact that high-dimensional arrays are used for the computation of the left ventricle volume during the isochoric phases. Further investigation on this issue is currently ongoing to enhance the computational performances even more.

The left ventricular pressures and volumes obtained using the FOM and the proposed Deep-HyROMnet for three parameter values are reported in Figures 5.9 and 5.10 for the coarser and the finer test cases, respectively, showing perfect agreement of the reduced outputs of interest with the high-fidelity ones, uniformly on the set of parameter inputs.

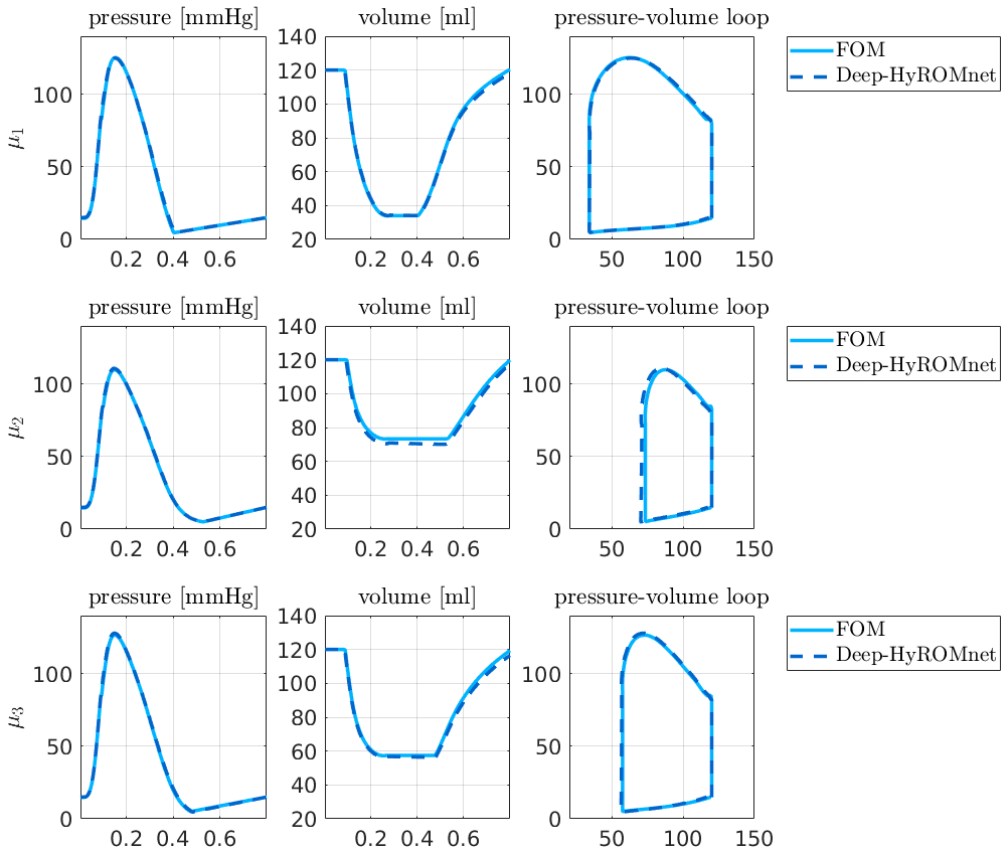


Figure 5.9: Cardiac cycle, physiological scenarios. Pressures, volumes and pressure-volume relationships, for $N_h = 18501$.

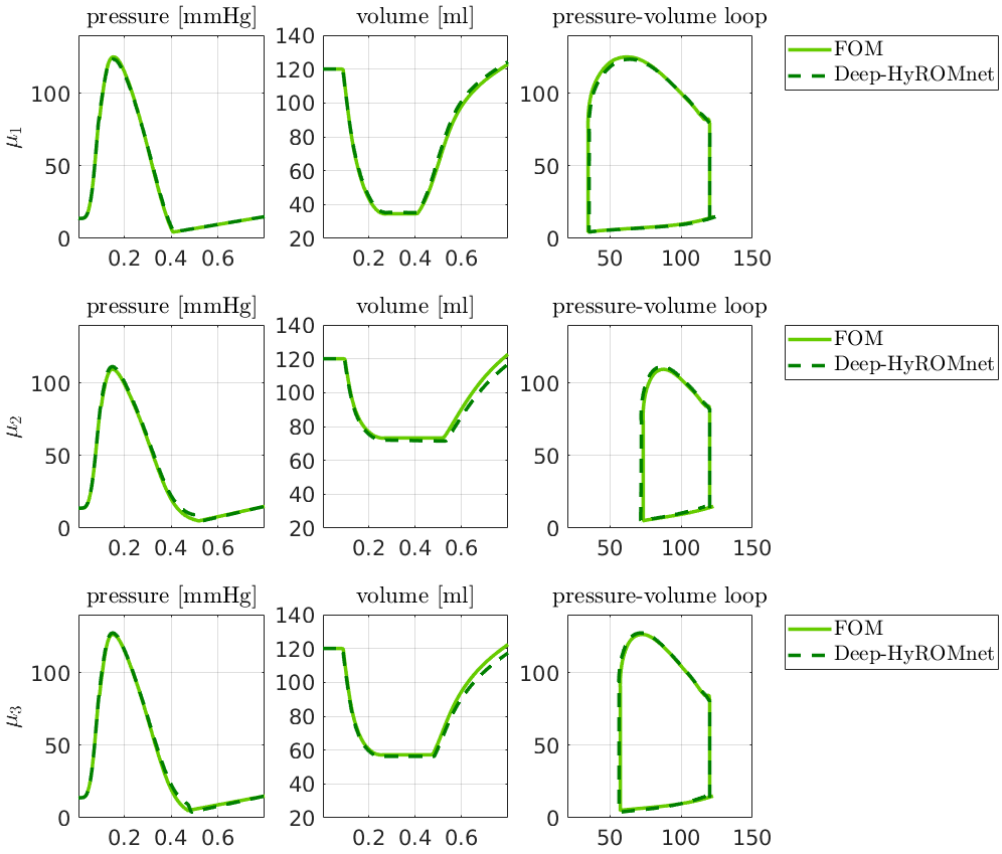


Figure 5.10: Cardiac cycle, physiological scenarios. Pressures, volumes and pressure-volume relationships, for $N_h = 126675$.

5.3.2 Pathological scenario

We now address the solution to the 3D-0D coupled problem in the eventuality that a portion of the cardiac tissue has been affected by myocardial ischemia, that is, a reduction of blood supply to the myocardium that may lead to the death of cells in the affected area [Gri+08]. In this case a reduced excitability of the cells and altered ionic currents are observed, as well as inhibited contractility of the tissue. For a detailed description and the numerical treatment of such pathological scenario from the electromechanical viewpoint we refer, e.g., to [Ger18; Sal+21]. Model order reduction in the case of ischemic necrosis is applied to the electrophysiology problem in [PMQ18; Fre+20]. However, ROMs have never been applied to characterized the mechanical behavior of the cardiac tissue in these scenarios for varying conditions of the ischemic tissue. This is the first time that such a phenomenon is investigated systematically in a broad variety of conditions.

In this chapter we exploit the Deep-HyROMnet technique to build a hyper-ROM for the mechanics of the left ventricle, when a region of the myocardium is affected by necrosis. In particular, let

$$B(\mathbf{X}_c, r) = \{\mathbf{X} \in \Omega_0 \mid \|\mathbf{X}_c - \mathbf{X}\|_2 < r\} \subset \mathbb{R}^3$$

be an idealized ischemic region with (fixed) center $\mathbf{X}_c \in \Omega_0$ and radius $r > 0$. To model the fact

that cardiomyocytes inside the necrosis behave as passive conductors, we assume zero active tension in the dofs belonging to the affected region. To be more specific, given τ_a^M as in (5.4), we define the following parameterized time-dependent function for the active stress in the fiber direction $\mathbf{T}_a(\mathbf{X}, t; \boldsymbol{\mu}) = T_a(\mathbf{X}, t; \boldsymbol{\mu})(\mathbf{F}\mathbf{f}_0 \otimes \mathbf{f}_0)$:

$$T_a(t; \boldsymbol{\mu}) = \frac{\widetilde{T}_a}{\max\limits_{t\in(0,0.8)s} \tau_a^M(t)} \tau_a^M\left(0.8 \left(\frac{t}{T} - \left\lfloor \frac{t}{T} \right\rfloor\right)\right) \chi_{B^c(\mathbf{X}_c, r)}(\mathbf{X}),$$

where we assume a uniform activation of the myocytes outside the scar $B(\mathbf{X}_c, r)$.

As unknown parameters for the pathological scenario, we choose

- the resistance of the windkessel model $R_p \in [2.5 \cdot 10^7, 4.5 \cdot 10^7]$ Pa $\cdot$ s $\cdot$ m^{-3},
- the active tension parameter $\widetilde{T}_a \in [4.5 \cdot 10^4, 6 \cdot 10^4]$ Pa,
- the radius of the ischemic region $r \in [0.2 \cdot 10^{-3}, 20 \cdot 10^{-3}]$ m,

that are the most influential parameters associated with the circulation model, the active component of the structural model (as seen in Chapter 3 for the benchmark of the passive inflation and the active contraction of an idealized left ventricle) and the necrotic region, respectively. Here we considered an hexahedral mesh with 6167 vertices, so that the FOM obtained using $\mathbb{Q}_1$-FE has dimension $N_h = 18501$.

Figure 5.11 shows the pressure-volume loops obtained for six different values of the vector $\boldsymbol{\mu} = [R_p, T_a, r] \in \mathcal{P}$, highlighting how parameter variations may have a great impact on clinical quantities of interest. In particular, we observe that the end systolic volume ranges from 40 ml to 72 ml, so that the ejection fraction (EF), that is the volumetric portion of blood ejected from the ventricle with each contraction, reduces from 67% to 40%. On the other hand, the maximum values of the blood pressure goes from 102 mmHg to almost 124 mmHg, thus influencing the slope of the end systolic pressure-volume relationship which provides an index

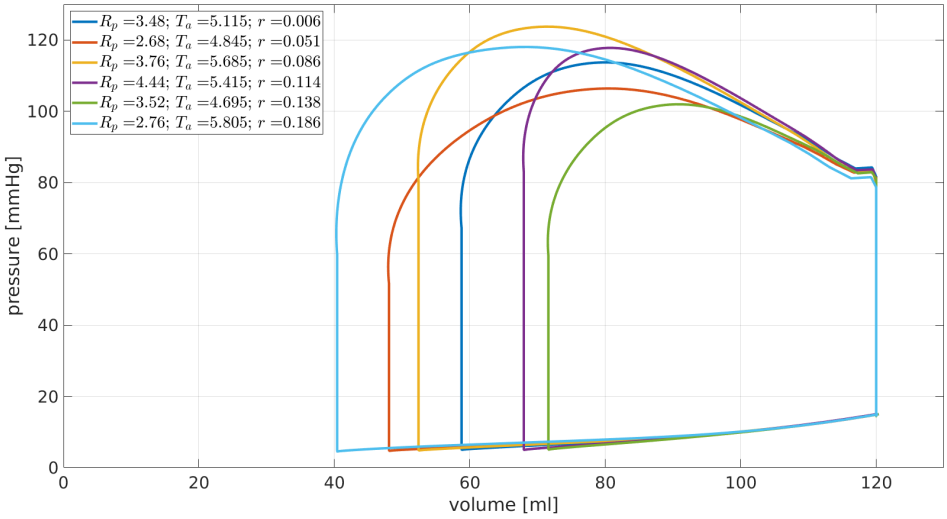


Figure 5.11: Cardiac cycle, pathological scenarios. Pressure-volume loops computed using the FOM with different parameter instances.

of myocardial contractility [Sat+98]. All these indicators are useful in clinical practice [Doy+13; BMS05; Sho15]. However, in order to gain more knowledge about the relationships between model inputs and clinical outputs of interest, sensitivity analysis studies should be performed.

The reduced basis is built by collecting high-fidelity solution snapshots for $n_s = 20$ parameter samples and performing randomized SVD using $N = 80$. In fact, a higher dimension of the RB basis with respect to the physiological scenario is required, possibly due to the presence of the ischemic region, as highlighted in Table 5.4. Indeed, the presence of the ischemic region ultimately makes the parameters-to-solution map more involved, affecting the behavior of the solution in a more pronounced way and thus requiring a higher dimension of the basis if a global linear subspace has to be used to approximate the whole solution manifold with sufficient accuracy.

POD tolerance ε_{POD}	10^{-3}	$5 \cdot 10^{-4}$	10^{-4}
RB dimension N	72	100	184

Table 5.4: Cardiac cycle, pathological scenarios. POD tolerances and associated RB dimension for the pathological scenario, when $N_h = 18501$.

Hence, $n_s' = 50$ ROM simulations are performed to collect the reduced nonlinear data (5.9) and the DNNs are trained. The results obtained using Deep-HyROMnet are reported in Table 5.5, where the average is computed over 20 testing parameters. As in the physiological scenario, Deep-HyROMnet requires less than 17 s to compute a whole heartbeat, so that it is almost 100 times faster than the FOM, which for the same task to be achieved requires almost 26 minutes. It is worth mentioning that relying on POD-Galerkin-DEIM ROMs led to negligible speed-ups (only 1.2 times faster that the FOM), thus making the development of the Deep-HyROMnet ROM necessary to efficiently address the solution to the problem under investigation. For what concerns the accuracy of the hyper-ROM, the absolute displacement error ϵ_{abs} is around $5 \cdot 10^{-2}$. Although more accurate results can be obtained with classical hyper-reduction techniques, a good approximation of the outputs of interest is obtained using Deep-HyROMnet. In particular, the error between the FOM and Deep-HyROMnet on the EF, computed over the testing set, is less than 3%. Figure 5.12 reports few examples of hyper-ROM displacement and pointwise error with respect to the FOM at time $t = 0.25$ s, when the ventricle in fully contracted, while the corresponding pressure-volume loops are shown in Figure 5.13.

	DEIM-3000	Deep-HyROMnet
Speed-up	$\times 1.2$	$\times 94$
Avg. CPU time	22 min	17 s
$\text{mean}_{\boldsymbol{\mu}}\ \epsilon_{abs}(\boldsymbol{\mu})$	$6 \cdot 10^{-3}$	$5 \cdot 10^{-2}$
$\text{mean}_{\boldsymbol{\mu}}\ \epsilon_{rel}(\boldsymbol{\mu})$	$2 \cdot 10^{-3}$	$1 \cdot 10^{-1}$

Table 5.5: Cardiac cycle, pathological scenarios. Computational data related to DEIM-based and DNN-based hyper-ROMs, for $N = 80$.

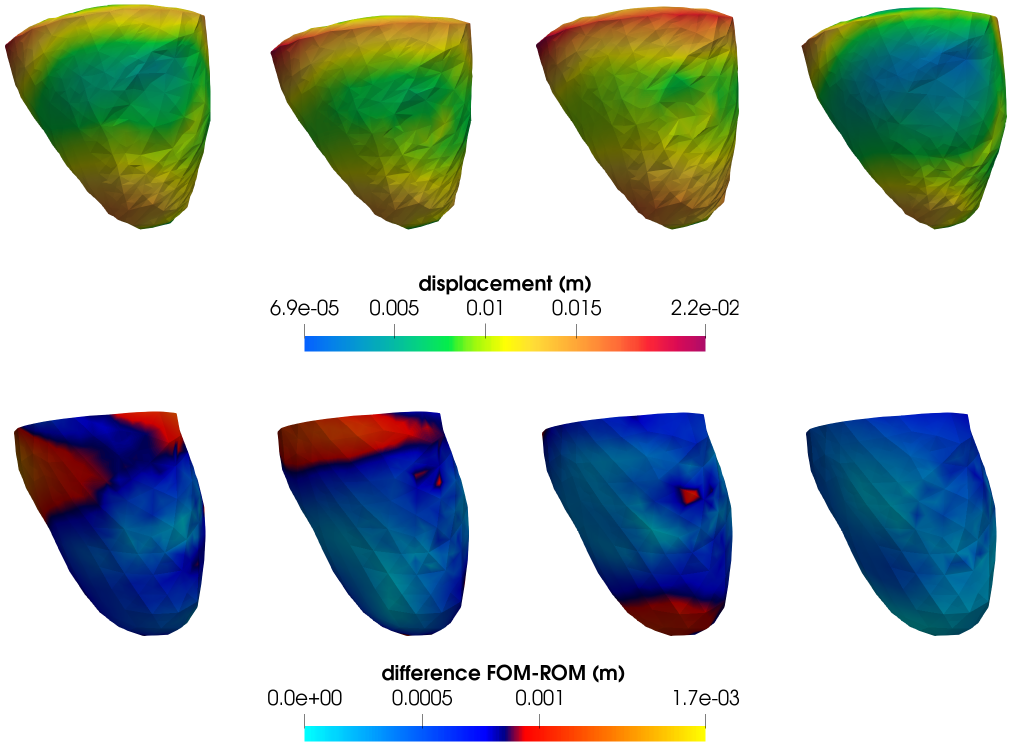


Figure 5.12: Cardiac cycle, pathological scenarios. Deep-HyROMnet deformation (top) and error (bottom) at time $t = 0.25$ s for different values of the parameter $\boldsymbol{\mu}$ (from left to right).

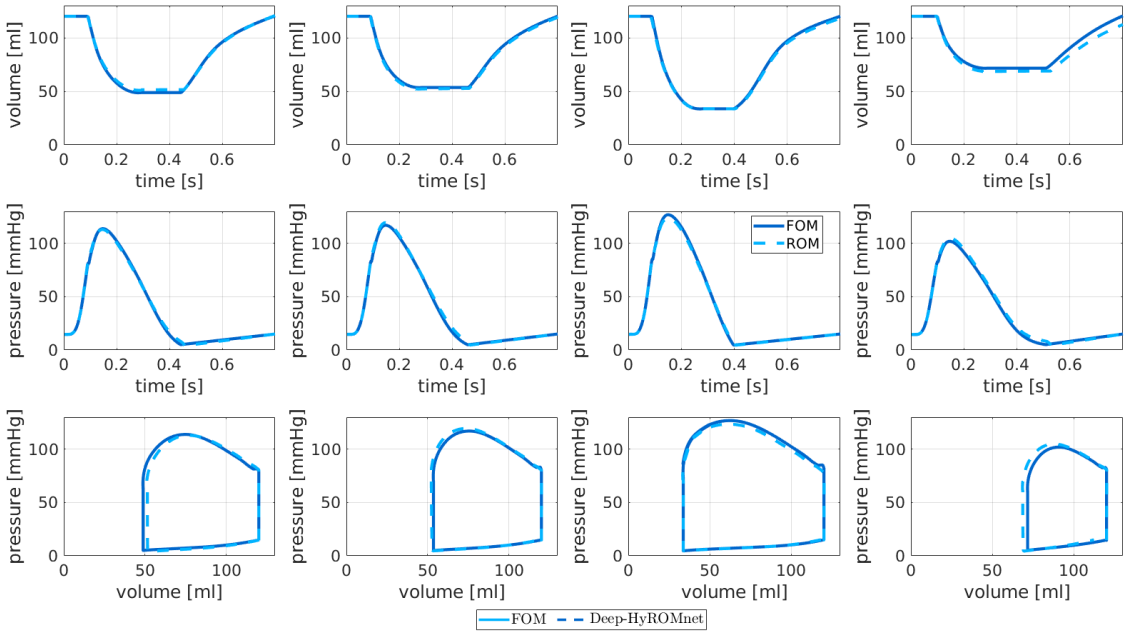


Figure 5.13: Cardiac cycle, pathological scenarios. Pressure, volumes and pressure-volume relationships for different values of the parameter $\boldsymbol{\mu}$ (from left to right).

5.3.3 Forward uncertainty quantification: preliminary results

To conclude, we address the repeated evaluation of the inputs-to-solution map in both physiological and pathological scenarios by means of the Deep-HyROMnet ROMs developed in Sections 5.3.1 and 5.3.2, in order to gain some useful knowledge about the impact of the model parameters on quantities of clinical interest.

With this aim, let us consider as model outputs:

- the ejection fraction, that represents the amount of blood pumped at each heartbeat, defined as
$$EF = \frac{EDV - ESV}{EDV},$$
where EDV and ESV denote the end-diastolic and the end-systolic volumes, respectively;
- the maximal rate of change in pressure
$$dP/dt_{max} = \max_{t\in(0,T)} \left(\frac{dp_{LV}(t;\boldsymbol{\mu})}{dt} \right) \approx \max_{n=1,\dots,N_t} \left(\frac{p_{LV}^n(\boldsymbol{\mu}) - p_{LV}^{n-1}(\boldsymbol{\mu})}{\Delta t} \right),$$
which is a common indicator of cardiac contractility.

These choice is motivated by the fact that both EF and dP/dt_{max} are commonly used mechanical biomarkers. Nonetheless, since Deep-HyROMnet computes the whole displacement at each time instance, any additional output, such as, e.g., the wall thickening, the end-systolic pressure or the longitudinal fractional shortening [LF+20; Cam+20], can be considered online without the need to rebuild the reduced order model. This is a distinguishing feature of the proposed reduction technique, compared to recent frameworks addressing NN-based approximation of output quantities of interest, without taking into account the approximation of the field variables involved in the output evaluations [Reg+21].

For what concerns the varying parameters, we consider:

- the resistance of the windkessel model $R_p \in [2.5 \cdot 10^7, 4.5 \cdot 10^7]$ Pa $\cdot$ s $\cdot$ m^{-3},
- the active tension parameter $\widetilde{T}_a \in [4.5 \cdot 10^4, 6 \cdot 10^4]$ Pa, and
- the radius of the ischemic region $r \in [0.2 \cdot 10^{-3}, 20 \cdot 10^{-3}]$ m,

so that we always assume $K = 5 \cdot 10^4$ Pa as online value for the bulk modulus in the healthy case as well.

The following results are obtained by performing 500 hyper-ROM simulations in physiological scenarios, i.e. for $\boldsymbol{\mu} = [R_p, \widetilde{T}_a]$, and 1000 in pathological ones, i.e. for $\boldsymbol{\mu} = [R_p, \widetilde{T}_a, r]$, taking into account an hexahedral computational mesh of a patient-specific left ventricle (see Figure 5.5) with 6167 vertices. The inputs are sampled over the parameter space using latin hypercube sampling. We recall that the underlying FOM dimension is $N_h = 18501$ and that less than 17 s are required by Deep-HyROMnet to compute the problem solution for each new instance of the parameter, thus entailing less than 7 hours of CPU time on a PC desktop computer with 3.70GHz Intel Core i5-9600K CPU and 16GB RAM. Performing these studies using the FOM would have required 27 days of computations, which become almost 240 days if the finer computational grid with 42225 vertices has to be considered (reducing to 38 hours when employing Deep-HyROMnet).

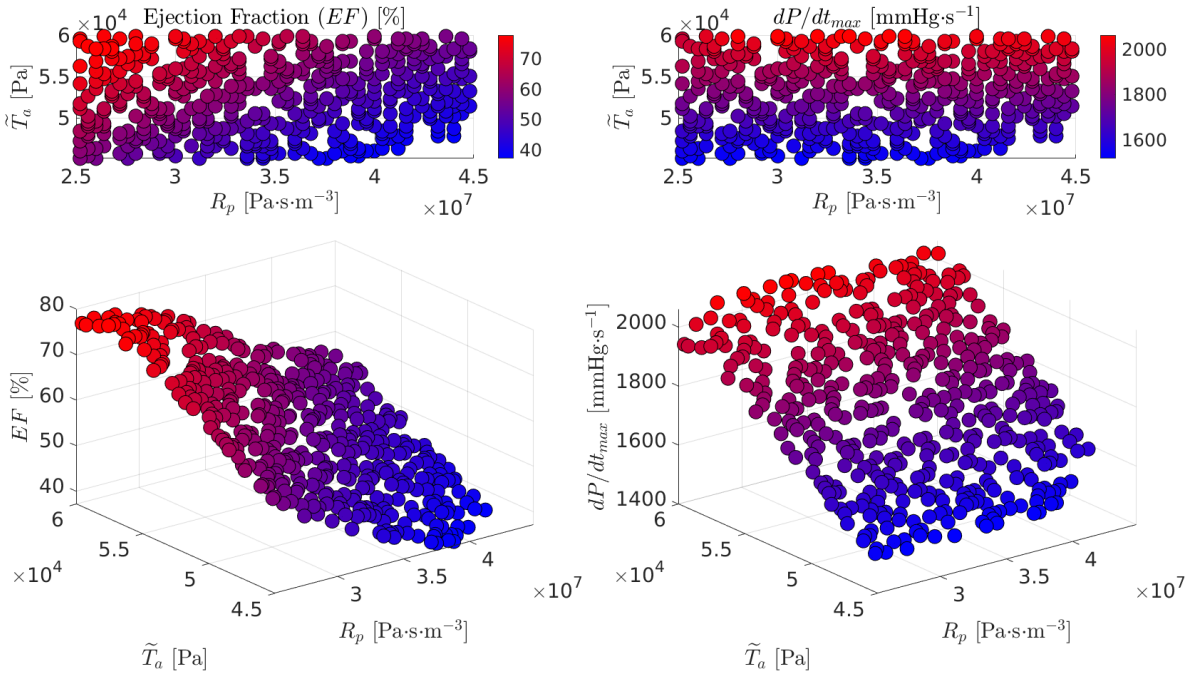
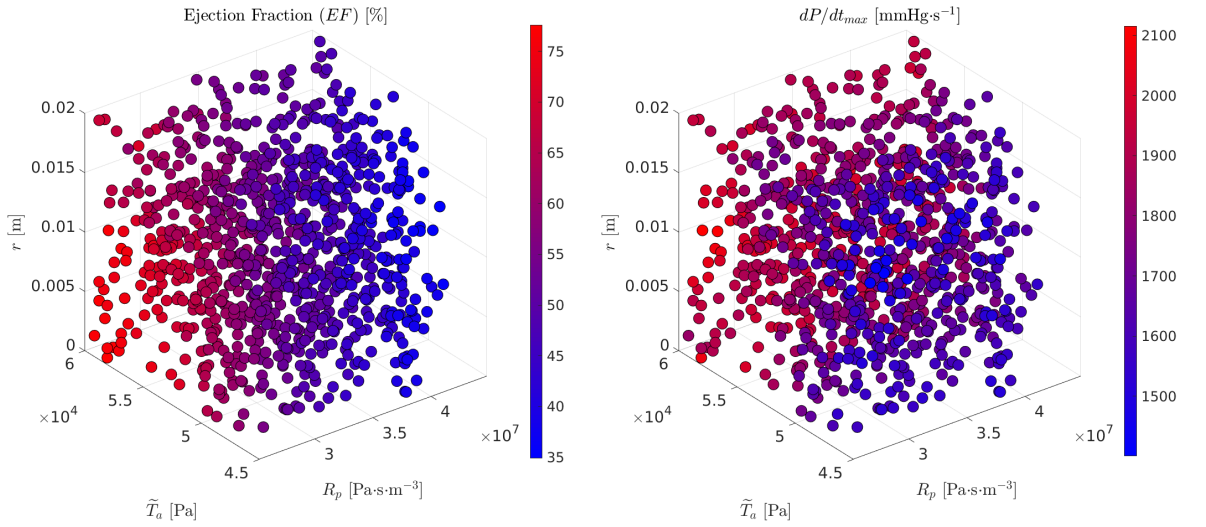


Figure 5.14: Cardiac cycle, physiological scenarios. Scatter plots of the EF (left) and the dP/dt_{max} (right) for 500 different parameters. The colormap represents the values of the output of interest.

Concerning the outcomes of the healthy scenarios, we observe that both the resistance R_p of the two-element windkessel model and the active stress $\widetilde{T}_a$ have a great impact on the EF, as we can see from the scatter plot reported in Figure 5.14 (left). Both parameters are, in fact, associated with the systolic phase of the cardiac cycle: larger values of the maximum active tension lead to a greater contraction of the myocardial tissue, whereas higher values of R_p correspond to a lower amount of blood that the ventricle is able to pump during ejection (phase 2 of the heartbeat, see Figure 5.2). As a consequence, they both affect the ESV without substantially changing the EDV. In particular, given a fixed value of R_p, the EF increases as $\widetilde{T}_a$ becomes higher; on the other way round, when $\widetilde{T}_a$ is fixed, the EF decreases as the resistance of the circulation model is increased. As an example, the minimum value $EF = 37\%$ corresponds to $\boldsymbol{\mu} = [4 \cdot 10^7 \text{ Pa} \cdot \text{s} \cdot \text{m}^{-3}, 4.605 \cdot 10^4 \text{ Pa}]$, that is, to a case where R_p and $\widetilde{T}_a$ are closed to their upper and lower bounds, respectively; its maximum value $EF = 78\%$ is obtained instead for $\boldsymbol{\mu} = [2.6 \cdot 10^7 \text{ Pa} \cdot \text{s} \cdot \text{m}^{-3}, 5.835 \cdot 10^4 \text{ Pa}]$. On the other hand, from the analysis of Figure 5.14 (right) we can conclude that the maximal rate of change in pressure is directly proportional to the active tension, going from 1763 $\text{mmHg} \cdot \text{s}^{-1}$ to 1998 $\text{mmHg} \cdot \text{s}^{-1}$ as $\widetilde{T}_a$ is increased from $4.515 \cdot 10^4$ Pa to its maximum value $5.985 \cdot 10^4$ Pa, whilst we observe that R_p has almost no influence on dP/dt_{max}.

Assessing the way input variations affect the considered outputs of interest in the pathological scenarios becomes more involved due to the presence of an additional parameter and to the fact that no activation of the cardiac myocytes is assumed inside the necrotic region $\mathcal{B}(\mathbf{X}_c, r)$, being $\mathbf{X}_c \in \Omega_0$ a fixed point inside the myocardium. In Figure 5.15 we report the scatter plots of EF and dP/dt_{max}, where in the x-,y- and z-axis are reported R_p, $\widetilde{T}_a$ and the radius r, respectively,



(a) EF **(b)** dP/dt_{max}

Figure 5.15: Cardiac cycle, pathological scenarios. Scatter plots of the EF (left) and the dP/dt_{max} (right) in the xyz-plane ($x = R_p$, $y = \widetilde{T}_a$ and $z = r$) for 1000 different parameters. The colormap represents the values of the outputs of interest.

while the colors of the data points encode the value of the outputs EF and dP/dt_{max}. Regarding the interaction between the maximum active tension $\widetilde{T}_a$ and the windkessel resistance R_p on their influence on the EF (Figure 5.16, left), we can draw similar conclusions to the healthy case. To give a few examples, the lower values of $34,9\%$ and 35.6% for the EF are computed for the parameters $\boldsymbol{\mu} = [4.48 \cdot 10^7 \text{ Pa} \cdot \text{s} \cdot \text{m}^{-3}, 4.905 \cdot 10^4 \text{ Pa}, 0.0118 \cdot \text{m}^{-3}]$ and $\boldsymbol{\mu} = [4.36 \cdot 10^7 \text{ Pa} \cdot \text{s} \cdot \text{m}^{-3}, 4.755 \cdot 10^4 \text{ Pa}, 0.0066 \cdot \text{m}^{-3}]$, respectively, whereas the higher values $EF = 76.8\%$ and 77.6% are associated with the inputs $\boldsymbol{\mu} = [2.60 \cdot 10^7 \text{ Pa} \cdot \text{s} \cdot \text{m}^{-3}, 5.835 \cdot 10^4 \text{ Pa}, 0.0018 \cdot \text{m}^{-3}]$ and $\boldsymbol{\mu} = [2.56 \cdot 10^7 \text{ Pa} \cdot \text{s} \cdot \text{m}^{-3}, 5.835 \cdot 10^4 \text{ Pa}, 0.011 \cdot \text{m}^{-3}]$. On the other hand, the influence of r on the EF is more difficult to ascertain from the analysis of the scatter plots.

In Figure 5.17 we report pressures and volumes for both the physiological and the pathological scenarios, obtained after one single heartbeat (so that the limit cycle is not reached). The colormaps refer to the size of the necrotic region, with $r = 0$ corresponding to a healthy heart. As we expect, the presence of the disease impacts the problem solution by reducing the ability of the ventricle to contract, resulting in higher volumes and lower pressures during ejection, corresponding to smaller values for the ESV with respect to the physiological scenarios, and thus smaller EF. Finally, from the 2D-views of the scatter plots reported in Figure 5.18, we can assume that variations of both R_p and r have almost no effect on the maximal rate of change of pressure dP/dt_{max}, and that $\widetilde{T}_a$ is the most influential parameter between those considered.

To conclude, we have observed that the maximum value of the active tension $\widetilde{T}_a$ has great influence on the EF and dP/dt_{max}, both in the physiological and in the pathological tests considered. This is in accordance with the results of sensitivity analysis conducted in [Cam+20] for the healthy left ventricle in a quasi-static assumption. Furthermore, the resistance R_p of the circulation model, associated with the ejection phase of ventricular systole, influences the values EF, as well as the size of the necrosis.

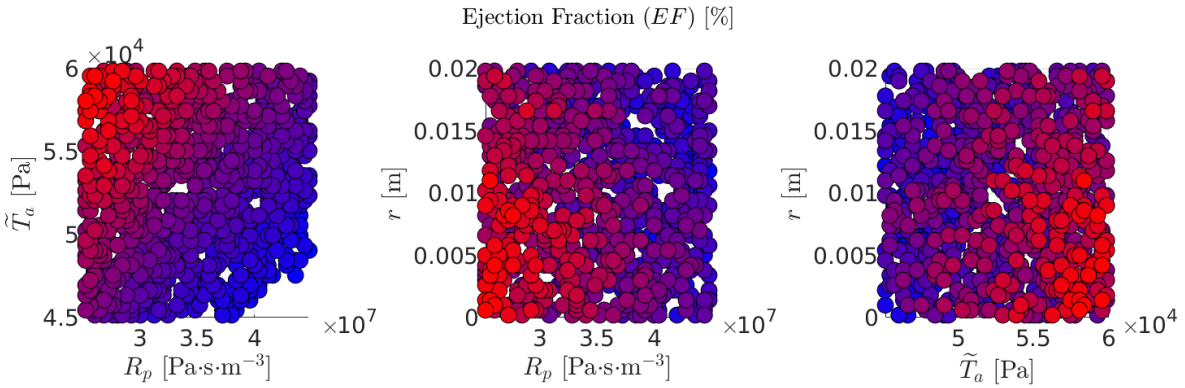


Figure 5.16: Cardiac cycle, pathological scenarios. 2D-views of the scatter plots of the EF for 1000 different parameters. The colormap represents the values of the output of interest.

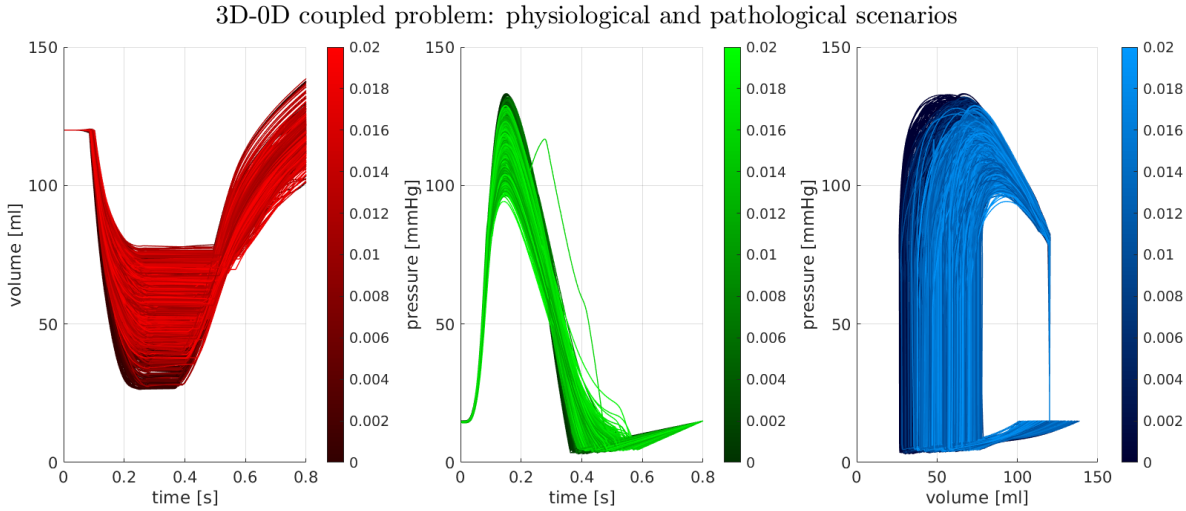


Figure 5.17: Cardiac cycle, physiological and pathological scenarios. Pressures (left), volumes (center) and pressure-volume loops (right) computed using the Deep-HyROMnet with 1500 different parameter instances. The colormap represents the values of the scar radius.

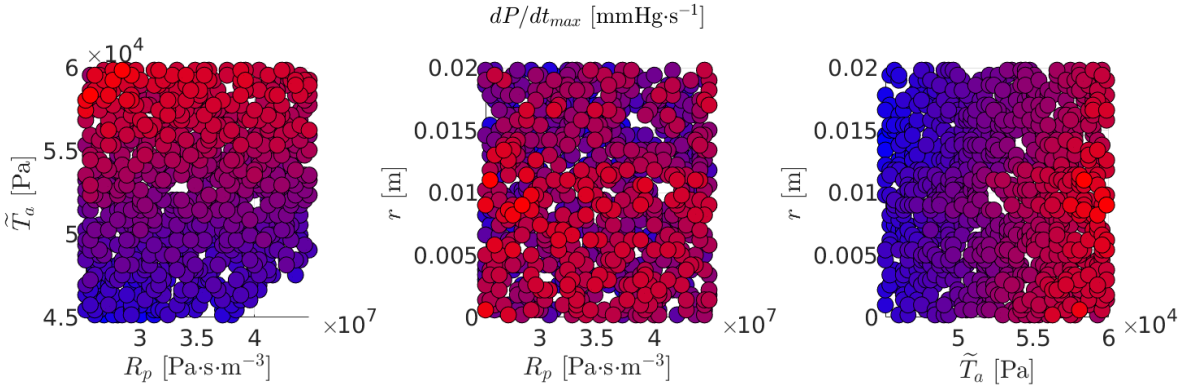


Figure 5.18: Cardiac cycle, pathological scenarios. 2D-views of the scatter plots of the dP/dt_{max} for 1000 different parameters. The colormap represents the values of the output of interest.

Conclusion

In this Thesis we have addressed the solution to nonlinear, time-dependent PDEs depending on several input parameters by means of suitable projection-based reduced order models (ROMs), developed to accurately capture the state solution dynamics at a reduced computational cost with respect to full order models (FOMs) providing expensive high-fidelity approximations. We focused in particular on the elastodynamics equation correlated with nonlinear constitutive laws, which can be used to describe the complex mechanical behavior of cardiac contraction and relaxation. Being able to evaluate the parameter-to-solution map in an inexpensive way, both in terms of CPU time and memory requirements, is essential to provide real-time solutions to cardiac mechanics models that can be carried out, e.g., on a laptop computer, thus providing an effective tool to embed numerical simulations of virtual hearts into the clinical practice.

In particular, the contributions of this Thesis are related with:

- the extension of state-of-art projection-based ROM strategies to tackle nonlinear parameterized PDEs in elastodynamics;
- the application of projection-based ROMs to sensitivity analysis and to the solution of uncertainty quantification problems in elastodynamics;
- the introduction of a new, deep learning-based strategy to perform hyper-reduction almost inexpensively and to overcome major computational bottlenecks in the construction of the ROMs;
- the application of the proposed strategies to cardiac mechanics problems, both in physiological and pathological scenarios.

We focused on Galerkin-reduced basis (RB) methods, characterized by a projection of the differential problem onto a low-dimensional subspace built, e.g., by performing proper orthogonal decomposition (POD) on a set of FOM solutions, and by the splitting of the reduction procedure into a costly offline phase and an inexpensive online phase. Numerical experiments showed that, despite their highly nonlinear nature, elastodynamics problems can be reduced by exploiting projection-based strategies in an effective way, with POD-Galerkin ROMs achieving very good accuracy even in presence of a handful of basis functions. However, when dealing with nonlinear problems, a further level of approximation is required to make the online stage independent of the high-fidelity dimension.

To efficiently handle the nonlinear operators, we considered the discrete empirical interpolation method (DEIM), properly combined with POD or randomized singular-value decomposition, as hyper-reduction technique. Examples of the performances of POD-Galerkin-DEIM ROMs in numerical tests reproducing different phases of the cardiac cycle, on both idealized and realistic geometries, have been shown. However, a serious issue is represented by the assembling (albeit onto a reduced mesh) of the approximated nonlinear operators in this framework, as a large DEIM basis is needed to ensure the convergence of Newton method for complex applications,

overall compromising the ROM efficiency. The highly nonlinear nature of the constitutive laws makes, in fact, the residual terms orders of magnitude more expensive to be approximated than the parameter-to-solution map. This observation suggested the idea of relying on surrogate models to perform operator approximation, overcoming the need to assemble the nonlinear terms onto the computational mesh.

Pursuing this strategy, we have proposed a new projection-based, deep learning-based ROM, *Deep-HyROMnet*, which combines the Galerkin-RB approach with deep neural networks (DNNs) to assemble the reduced Newton system in an efficient way, thus avoiding the computational burden entailed by classical hyper-reduction strategies. This approach allows to rely on physics-based (thus, consistent) ROMs retaining the underlying structure of the physical model, as DNNs are employed only for the approximation of the reduced nonlinear operators, so that the problem displacement at each time instance is computed by solving the reduced nonlinear system. Regarding the offline cost of this hybrid reduction strategy, we point out that:

- FOM solutions are required only for the construction of the reduced basis functions;
- since the nonlinear operators are collected during Newton iterations at each time step, a smaller number of ROM simulations with respect to purely data-driven approaches is sufficient for training the DNNs;
- being the training data low-dimensional, the overwhelming training times and costs that would be required by the DNN if FOM array were used can be avoided.

Sensitivity analysis (SA) carried out on structural problems showed that the FOM can be replaced by more efficient ROMs without affecting the results of the studies. On the other way round, being able to quantify the impact of parameter variations on outputs associated with the problem solution can be extremely informative for the construction of the ROMs themselves, as the number of varying input factors can be narrowed to the most influential ones. This fact is particularly useful in the construction of Deep-HyROMnet ROMs, since it allows to reduced the size of the network inputs, which depends on the number of model parameters, as well as the dimension of the training data, i.e. the reduced nonlinear operators, ultimately depending on the dimension of the RB space. Another important aspect that has emerged is that the choice of the quantities of interest affects the outcome of the SA, meaning that some factors having a great impact on a specific output may be less influential for others. Considering the (idealized) ventricle systole, the active tension showed to have a greater impact on the contraction of the ventricle, whereas the fiber orientation controls the typical twisting motion. In view of this, relying on physics-based ROMs, providing an approximation at each time step of the displacement vector, allow to consider new quantities of interest without the need to repeat the whole reduction procedure or the training process.

To conclude, we have shown how Deep-HyROMnets outperform POD-Galerkin-DEIM ROMs in terms of computational speed-up and allow to address the efficient solution to cardiac mechanics problems coupled with a lumped-parameter model for blood circulation, for which classical hyper-ROMs require high computational costs. Preliminary results of forward uncertainty quantification carried out on physiological and pathological test cases using a patient-specific left ventricle allowed to gain some useful knowledge about the impact of the model parameters on quantities of clinical interest. In particular, we have observed that the active tension has great influence on both the considered clinical biomarkers, that are the ejection fraction and

the maximal rate of change in pressure, while other parameters, such as the resistance of the circulation model and the size of the necrosis, showed higher influence on the ejection fraction only.

By providing a reliable and computationally efficient reduction procedure, Deep-HyROMnet can be successfully used to address the solution to multy-query problems, such as parameter estimation and uncertainty quantification. Nonetheless, a number of issues still need to be further investigated. First of all, the assessment of the reduction strategy on even more involved scenarios, among which we mention:

- finer computational meshes, biventricular geometries, smaller time steps;
- a surrogate model for the computation of space- and time-dependent active tension, thus taking into account the activation of cardiac myocytes at different time instants;
- more and/or different input parameters according to sensitivity analysis carried on the simulation of the whole cardiac cycle.

Another aspect that should be addressed is the derivation of a posteriori error estimator accounting for the errors related to the solution-space reduction and to the hyper-reduced approximation of the nonlinear terms, at least when relying on POD-Galerkin-DEIM ROMs. In addition, the investigation of Deep-HyROMnet approximation properties when deeper neural network architectures are employed or a larger number of online inputs (e.g. including the reduced Newton increments) are considered can enhance the development of more accurate hyper-reduced models. Finally, a multi-fidelity approach, combining the FOM, POD-Galerkin-DEIM ROMs and Deep-HyROMnets, as well as other low-fidelity ROMs, could be beneficial in order to face larger scale applications.

Appendix

Cardiac mechanics simulation parameters

The reference parameters for the 3D-0D structure-windkessel model used in this Thesis are summarized in this appendix.

Name	**Parameter**	**Value**	**Unit**
Cardiac mechanics			
Tissue density	ρ_0	10^3	$\mathrm{kg} \cdot \mathrm{m}^{-3}$
Robin boundary condition	$K_\perp$	$2 \cdot 10^5$	$\mathrm{Pa} \cdot \mathrm{m}^{-1}$
Robin boundary condition	$K_\parallel$	$2 \cdot 10^4$	$\mathrm{Pa} \cdot \mathrm{m}^{-1}$
Robin boundary condition	$C_\perp$	$2 \cdot 10^4$	$\mathrm{Pa} \cdot \mathrm{s} \cdot \mathrm{m}^{-1}$
Robin boundary condition	$C_\parallel$	$2 \cdot 10^3$	$\mathrm{Pa} \cdot \mathrm{s} \cdot \mathrm{m}^{-1}$
Passive myocardial tissue			
Hyperelastic parameter	b_f	8	
Hyperelastic parameter	b_s	6	
Hyperelastic parameter	b_n	3	
Hyperelastic parameter	b_{fs}	12	
Hyperelastic parameter	b_{fn}, b_{sn}	3	
Material stiffness	C	880	Pa
Bulk modulus	K	$5 \cdot 10^4$	Pa
Active myocardial tissue			
Maximum active tension	$\widetilde{T}_a$	$5 \cdot 10^4$	Pa
Fiber angle	$\boldsymbol{\alpha}^{epi}$	-60	deg
Fiber angle	$\boldsymbol{\alpha}^{endo}$	60	deg
Fiber angle	$\boldsymbol{\beta}^{epi}$	20	deg
Fiber angle	$\boldsymbol{\beta}^{endo}$	-20	deg

Circulation			
Capacitance	C_p	$4.5 \cdot 10^9$	$\mathrm{m}^{-3} \cdot \mathrm{Pa}^{-1}$
Resistance	R_p	$3.5 \cdot 10^7$	$\mathrm{Pa} \cdot \mathrm{s} \cdot \mathrm{m}^{-3}$
End-diastolic pressure	p_{ED}	15	mmHg
Aortic valve opening pressure	p_{AVO}	$82,50$	mmHg
Mitral valve opening pressure	p_{MVO}	5	mmHg

Table .6: Reference values of the input parameters to the 3D-0D problem.

Bibliography

[Aba+16] M Abadi et al. “Tensorflow: A system for large-scale machine learning”. In: *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*. 2016, pp. 265–283.

[Ada+20] B Adams, W Bohnhoff, K Dalbey, M Ebeida, J Eddy, M Eldred, R Hooper, P Hough, K Hu, J Jakeman, M Khalil, K Maupin, JA Monschke, E Ridgway, A Rushdi, D Seidl, J Stephens, and J Winokur. *Dakota, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 6.13 User’s Manual.* Tech. rep. Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2020.

[AP12] D Ambrosi and S Pezzuto. “Active stress vs. active strain in mechanobiology: constitutive issues”. In: *Journal of Elasticity* 107.2 (2012), pp. 199–212.

[AZF12] D Amsallem, MJ Zahr, and C Farhat. “Nonlinear model order reduction based on local reduced-order bases”. In: *Int. J. Numer. Meth. Engng.* 92.10 (2012), pp. 891–916.

[Arn+20] D Arndt, W Bangerth, B Blais, TC Clevenger, M Fehling, AV Grayver, T Heister, L Heltai, M Kronbichler, M Maier, P Munch, JP Pelteret, R Rastak, I Thomas, B Turcksin, Z Wang, and D Wells. “The `deal.II` Library, Version 9.2”. In: *Journal of Numerical Mathematics* 28.3 (2020), pp. 131–146.

[Ast+08] P Astrid, S Weiland, K Willcox, and T Backx. “Missing point estimation in models described by proper orthogonal decomposition”. In: *IEEE Transactions on Automatic Control* 53.10 (2008), pp. 2237–2251.

[Aub91] N Aubry. “On the hidden beauty of the proper orthogonal decomposition”. In: *Theoretical and Computational Fluid Dynamics* 2.5 (1991), pp. 339–352.

[Aug+16] CM Augustin, A Neic, M Liebmann, AJ Prassl, SA Niederer, G Haase, and G Plank. “Anatomically accurate high resolution modeling of human whole heart electromechanics: a strongly scalable algebraic multigrid solver method for nonlinear deformation”. In: *Journal of computational physics* 305 (2016), pp. 622–646.

[Bar+18] L Barbarotta, S Rossi, L Dede’, and A Quarteroni. “A transmurally heterogeneous orthotropic activation model for ventricular contraction and its numerical validation”. In: *Int. J. Numer. Meth. Biomed. Engng.* 34.12 (2018), e3137.

[Bar+04] M Barrault, Y Maday, NC Nguyen, and AT Patera. “An ‘empirical interpolation’ method: application to efficient reduced-basis discretization of partial differential equations”. In: *Comptes Rendus Mathematique* 339.9 (2004), pp. 667–672.

[Bay+12] JD Bayer, RC Blake, G Plank, and NA Trayanova. “A novel rule-based algorithm for assigning myocardial fiber orientation to computational heart models”. In: *Annals of biomedical engineering* 40.10 (2012), pp. 2243–2254.

[Ben+20] P Benner, P Goyal, B Kramer, B Peherstorfer, and K Willcox. "Operator inference for non-intrusive model reduction of systems with non-polynomial nonlinear terms". In: *Computer Methods in Applied Mechanics and Engineering* 372 (2020), p. 113433.

[BGW15] P Benner, S Gugercin, and K Willcox. "A survey of projection-based model reduction methods for parametric dynamical systems". In: *SIAM review* 57.4 (2015), pp. 483–531.

[Ben+17] P Benner, M Ohlberger, A Patera, G Rozza, and K Urban. *Model reduction of parametrized systems*. Springer, 2017.

[BGL05] M Benzi, GH Golub, and J Liesen. "Numerical solution of saddle point problems". In: *Acta numerica* 14 (2005), pp. 1–137.

[Ber02] DM Bers. "Cardiac excitation–contraction coupling". In: *Nature* 415.6868 (2002), pp. 198–205.

[Ber01] Donald Bers. *Excitation-contraction coupling and cardiac contractile force*. Vol. 237. Springer Science & Business Media, 2001.

[Bha+21] K Bhattacharya, B Hosseini, NB Kovachki, and AM Stuart. "Model reduction and neural networks for parametric PDEs". In: *The SMAI journal of computational mathematics* 7 (2021), pp. 121–157.

[BF10] PJ Blanco and RA Feijóo. "A 3D-1D-0D Computational model for the entire cardiovascular system". In: *Mecánica Computacional* 29.59 (2010), pp. 5887–5911.

[BW97] J Bonet and RD Wood. *Nonlinear continuum mechanics for finite element analysis*. Cambridge university press, 1997.

[BMQ17] D Bonomi, A Manzoni, and A Quarteroni. "A matrix DEIM technique for model reduction of nonlinear parametrized problems in cardiac mechanics". In: *Comput. Meth. Appl. Mech. Engng.* 324 (2017), pp. 300–326.

[BMQ16] D Bonomi, A Manzoni, and A Quarteroni. "A matrix discrete empirical interpolation method for the efficient model reduction of parametrized nonlinear PDEs: Application to nonlinear elasticity problems". In: *MATHICSE, Ecole Polytechn. Fédérale de Lausanne, Lausanne, Switzerland, Tech. Rep* 14 (2016), p. 2016.

[Bro65] CG Broyden. "A Class of Methods for Solving Nonlinear Simultaneous Equations". In: *Mathematics of Computation* 19.92 (1965), pp. 577–593. ISSN: 00255718, 10886842.

[BMS05] D Burkhoff, I Mirsky, and H Suga. "Assessment of systolic and diastolic ventricular properties via pressure-volume analysis: a guide for clinical, translational, and basic researchers". In: *American Journal of Physiology-Heart and Circulatory Physiology* 289.2 (2005), H501–H512.

[Cam+20] JO Campos, J Sundnes, RW Dos Santos, and BM Rocha. "Uncertainty quantification and sensitivity analysis of left ventricular function during the full cardiac cycle". In: *Philosophical Transactions of the Royal Society A* 378.2173 (2020), p. 20190381.

[CBMF11] K Carlberg, C Bou-Mosleh, and C Farhat. “Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations”. In: *International Journal of Numerical Methods in Engineering* 86.2 (2011), pp. 155–181.

[CTB15] K Carlberg, R Tuminaro, and P Boggs. “Preserving Lagrangian structure in nonlinear model reduction with application to structural dynamics”. In: *SIAM Journal on Scientific Computing* 37.2 (2015), B153–B184.

[Cha+16] R Chabiniok, VY Wang, M Hadjicharalambous, L Asner, J Lee, M Sermesant, E Kuhl, AA Young, P Moireau, MP Nash, D Chapelle, and DA Nordsletten. “Multiphysics and multiscale modelling, data–model fusion and integration of organ physiology in the clinic: ventricular cardiac mechanics”. In: *Interface focus* 6.2 (2016), p. 20150083.

[Cha+13] D Chapelle, A Gariah, P Moireau, and J Sainte-Marie. “A Galerkin strategy with Proper Orthogonal Decomposition for parameter-dependent problems – Analysis, assessments and applications to parameter estimation”. In: *ESAIM: Math. Mod. Num. Anal.* 47.6 (2013), 1821–1843.

[CS10] S Chaturantabut and DC Sorensen. “Nonlinear model reduction via discrete empirical interpolation”. In: *SIAM Journal on Scientific Computing* 32.5 (2010), pp. 2737–2764.

[CC95] T Chen and H Chen. “Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems”. In: *IEEE Transactions on Neural Networks* 6.4 (1995), pp. 911–917.

[Che05] Z Chen. *Finite Element Methods and Their Applications*. Springer Science & Business Media, 2005.

[Cia02] PG Ciarlet. *The finite element method for elliptic problems*. SIAM, 2002.

[CHM01] KD Costa, JW Holmes, and AD McCulloch. “Modelling cardiac mechanical properties in three dimensions”. In: *Philosophical transactions of the Royal Society of London. Series A: Mathematical, physical and engineering sciences* 359.1783 (2001), pp. 1233–1250.

[DGQ20] L Dede’, A Gerbi, and A Quarteroni. “Segregated algorithms for the numerical simulation of cardiac electromechanics in the left human ventricle”. In: *The Mathematics of Mechanobiology*. Springer, 2020, pp. 81–116.

[Dem81] H Demiray. “Large deformation analysis of some soft biological tissues”. In: (1981).

[DS00] S Doll and K Schweizerhof. “On the development of volumetric strain energy functions”. In: *J. Appl. Mech.* 67.1 (2000), pp. 17–21.

[Dos+19] R Doste, D Soto-Iglesias, G Bernardino, A Alcaine, R Sebastian, S Giffard-Roisin, M Sermesant, A Berruezo, D Sanchez-Quintana, and O Camara. “A rule-based method to model myocardial fiber orientation in cardiac biventricular geometries with outflow tracts”. In: *Int. J. Numer. Meth. Biomed. Engng.* 35.4 (2019), e3185.

[Doy+13] M Doyle et al. “Left ventricular energy model predicts adverse events in women with suspected myocardial ischemia: results from the NHLBI-sponsored women’s ischemia syndrome evaluation (WISE) study”. In: *Cardiovascular diagnosis and therapy* 3.2 (2013), p. 64.

[DHO12] M Drohmann, B Haasdonk, and M Ohlberger. “Reduced basis approximation for nonlinear parametrized evolution equations based on empirical operator interpolation”. In: *SIAM Journal on Scientific Computing* 34.2 (2012), A937–A969.

[EY36] C Eckart and G Young. “The approximation of one matrix by another of lower rank”. In: *Psychometrika* 1.3 (1936), pp. 211–218.

[Edm71] J Edmonds. “Matroids and the greedy algorithm”. In: *Mathematical programming* 1.1 (1971), pp. 127–136.

[EG13] A Ern and JL Guermond. *Theory and practice of finite elements*. Vol. 159. Springer Science & Business Media, 2013.

[ES19] S Eugster and D Steigmann. “Continuum theory for mechanical metamaterials with a cubic lattice substructure”. In: *Mathematics and Mechanics of Complex Systems* 7.1 (2019), pp. 75–98.

[Far+20] C Farhat, S Grimberg, A Manzoni, and A Quarteroni. “Computational bottlenecks for PROMs: precomputation and hyperreduction”. In: *P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. Schilders, L. Silveira (eds.) Model Order Reduction. Volume 2: Snapshot-Based Methods and Algorithms*. De Gruyter, Berlin, (2020, in press), pp. 181–244.

[FQ21] M Fedele and A Quarteroni. “Polygonal surface processing and mesh generation tools for the numerical simulation of the cardiac function”. In: *Int. J. Numer. Meth. Biomed. Engng.* 37.4 (2021), e3435.

[FDM21] S Fresca, L Dede’, and A Manzoni. “A comprehensive deep learning-based approach to reduced order modeling of nonlinear time-dependent parametrized PDEs”. In: *Journal of Scientific Computing* 87.2 (2021), pp. 1–36.

[Fre+20] S Fresca, A Manzoni, L Dede’, and A Quarteroni. “Deep learning-based reduced order models in cardiac electrophysiology”. In: *PloS one* 15.10 (2020), e0239416.

[FM22] Stefania Fresca and Andrea Manzoni. “POD-DL-ROM: enhancing deep learning-based reduced order models for nonlinear parametrized PDEs by proper orthogonal decomposition”. In: *Computer Methods in Applied Mechanics and Engineering* 388 (2022), p. 114181.

[GWZ20] H Gao, JX Wang, and MJ Zahr. “Non-intrusive model reduction of large-scale, nonlinear dynamical systems using deep learning”. In: *Physica D: Nonlinear Phenomena* 412 (2020), p. 132614.

[GG97] A Gelman and WR Gilks. “Weak convergence and optimal scaling of random walk Metropolis algorithms’”. In: *The Annals of Applied Probability* 7.1 (1997), pp. 110–120.

[Ger18] A Gerbi. *Numerical approximation of cardiac electro-fluid-mechanical models*. Tech. rep. EPFL, 2018.

[GDQ19] A Gerbi, L Dede', and A Quarteroni. "A monolithic algorithm for the simulation of cardiac electromechanics in the human left ventricle". In: (2019).

[GTS17] F Ghavamian, P Tiso, and A Simone. "POD–DEIM model order reduction for strain-softening viscoplasticity". In: *Computer Methods in Applied Mechanics and Engineering* 317 (2017), pp. 458–479.

[GBC16] I Goodfellow, Y Bengio, and A Courville. *Deep Learning*. MIT Press, 2016.

[Gre+81] RA Greenbaum, SY Ho, DG Gibson, AE Becker, and RH Anderson. "Left ventricular fibre architecture in man". In: *Heart* 45.3 (1981), pp. 248–263.

[Gre+07] MA Grepl, Y Maday, NC Nguyen, and AT Patera. "Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations". In: *ESAIM: Mathematical Modelling and Numerical Analysis* 41.3 (2007), pp. 575–605.

[Gri+08] BP Griffin, EJ Topol, D Nair, and K Ashley. *Manual of cardiovascular medicine*. Lippincott Williams & Wilkins, 2008.

[GCM95] JM Guccione, KD Costa, and AD McCulloch. "Finite element stress analysis of left ventricular mechanics in the beating dog heart". In: *Journal of biomechanics* 28.10 (1995), pp. 1167–1177.

[GM91] JM Guccione and AD McCulloch. "Finite element modeling of ventricular mechanics". In: *Theory of Heart*. Springer, 1991, pp. 121–144.

[GM93] JM Guccione and AD McCulloch. "Mechanics of active contraction in cardiac muscle: Part I–Constitutive relations for fiber stress that describe deactivation." In: *Journal of Biomechanical Engineering* 115.1 (1993), pp. 72–81.

[HMT11] N Halko, PG Martinsson, and JA Tropp. "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions". In: *SIAM review* 53.2 (2011), pp. 217–288.

[Hau+18] P Hauseux, JS Hale, S Cotin, and SPA Bordas. "Quantifying the uncertainty in a hyperelastic soft tissue model with stochastic parameters". In: *Appl. Math. Model.* 62 (2018), pp. 86–102.

[HRS16] JS Hesthaven, G Rozza, and B Stamm. *Certified reduced basis methods for parametrized partial differential equations*. Vol. 590. Springer, 2016.

[HSZ14] JS Hesthaven, B Stamm, and S Zhang. "Efficient greedy algorithms for high-dimensional parameter spaces with applications to empirical interpolation and reduced basis methods". In: *ESAIM Mathematical Modelling and Numerical Analysis* 48 (Jan. 2014).

[HU18] JS Hesthaven and S Ubbiali. "Non-intrusive reduced order modeling of nonlinear problems using neural networks". In: *Journal of Computational Physics* 363 (2018), pp. 55–78.

[Hij+20] S Hijazi, G Stabile, A Mola, and G Rozza. "Data-driven POD-Galerkin reduced order model for turbulent flows". In: *Journal of Computational Physics* 416 (2020), p. 109513.

[Hir18] M Hirschvogel. "Computational modeling of patient-specific cardiac mechanics with model reduction-based parameter estimation and applications to novel heart assist technologies". PhD thesis. Technische Universität München, 2018.

[Hir+17] M Hirschvogel, M Bassilious, L Jagschies, SM Wildhirt, and MW Gee. "A monolithic 3d-0d coupled closed-loop model of the heart and the vascular system: experiment-based parameter estimation for patient-specific cardiac mechanics". In: *International Journal for Numerical Methods in Biomedical Engineering* 33.8 (2017), e2842.

[Hol00] GA Holzapfel. *Nonlinear Solid Mechanics. A Continuum Approach for Engineering*. Wiley, Chichester, 2000.

[HO09] GA Holzapfel and RW Ogden. "Constitutive modelling of passive myocardium: a structurally based framework for material characterization". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 367.1902 (2009), pp. 3445–3475.

[HS96] T Homma and A Saltelli. "Importance measures in global sensitivity analysis of nonlinear models". In: *Reliability Engineering & System Safety* 52.1 (1996), pp. 1–17.

[HCM17] DE Hurtado, S Castro, and P Madrid. "Uncertainty quantification of 2 models of cardiac electromechanics". In: *Int. J. Numer. Meth. Biomed. Engng.* 33.12 (2017), e2894.

[Iai09] PA Iaizzo. *Handbook of Cardiac Anatomy, Physiology, and Devices*. Springer Science & Business Media, 2009.

[Inc14] Z. M. G. Inc. "Zygote Solid 3D heart Generation II Development Report". In: *Technical Report* (2014).

[JKT06] G Jenkins, C Kemnitz, and GJ Tortora. *Anatomy and Physiology: From Science to Life*. Wiley, 2006. ISBN: 9780471613183.

[Kas+89] DA Kass, R Beyar, E Lankford, M Heard, WL Maughan, and K Sagawa. "Influence of contractile state on curvilinearity of in situ end-systolic pressure-volume relations." In: *Circulation* 79.1 (1989), pp. 167–178.

[Kat10] AM Katz. *Physiology of the Heart*. Lippincott Williams & Wilkins, 2010.

[Klo+06] S Klotz, I Hay, ML Dickstein, GH Yi, J Wang, MS Maurer, DA Kass, and D Burkhoff. "Single-beat estimation of end-diastolic pressure-volume relationship: a novel method with potential for noninvasive application". In: *American Journal of Physiology-Heart and Circulatory Physiology* 291.1 (2006), H403–H412.

[Kov+21] E Kovacheva, L Thämer, T Fritz, G Seemann, M Ochs, O Dössel, and A Loewe. "Estimating cardiac active tension from wall motion—An inverse problem of cardiac biomechanics". In: *International Journal for Numerical Methods in Biomedical Engineering* (2021), e3448.

[Lan+15] S Land et al. "Verification of cardiac mechanics software: benchmark problems and solutions for testing active and passive material behaviour". In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 471.2184 (2015), p. 20150641.

[LF+20] F Levrero-Florencio, F Margara, E Zacur, A Bueno-Orovio, ZJ Wang, A Santiago, J Aguado-Sierra, G Houzeaux, V Grau, D Kay, M Vázquez, R Ruiz-Baier, and B Rodriguez. "Sensitivity analysis of a strongly-coupled human-based electromechanical cardiac model: Effect of mechanical parameters on physiologically relevant biomarkers". In: *Computer methods in applied mechanics and engineering* 361 (2020), p. 112762.

[Lia+02] YC Liang, HP Lee, SP Lim, WZ Lin, KH Lee, and CG Wu. "Proper orthogonal decomposition and its applications—Part I: Theory". In: *Journal of Sound and vibration* 252.3 (2002), pp. 527–544.

[LY98] DHS Lin and FCP Yin. "A multiaxial constitutive law for mammalian left ventricular myocardium in steady-state barium contracture or tetanus". In: *Journal of biomechanical engineering* 120.4 (1998), pp. 504–517.

[Lu+21] L Lu, P Jin, G Pang, Z Zhang, and GE Karniadakis. "Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators". In: *Nature Machine Intelligence* 3.3 (2021), pp. 218–229.

[Lum67] JL Lumley. "The structure of inhomogeneous turbulent flows". In: *Atmospheric turbulence and radio wave propagation* (1967).

[MBQ18] A Manzoni, D Bonomi, and A Quarteroni. "Reduced order modeling for cardiac electrophysiology and mechanics: New methodologies, challenges and perspectives". In: *Mathematical and Numerical Modeling of the Cardiovascular System and Applications*. Ed. by D. Boffi, L. Pavarino, G. Rozza, S. Scacchi, and C. Vergara. Vol. 16. SEMA SIMAI Springer Series. Springer, Cham, 2018, pp. 115–166.

[MPL16] A Manzoni, S Pagani, and T Lassila. "Accurate solution of Bayesian inverse uncertainty quantification problems combining reduced basis methods and reduction error models". In: *SIAM/ASA Journal on Uncertainty Quantification* 4.1 (2016), pp. 380–412.

[MS07] LG Maqueda and AA Shabana. "Poisson modes and general nonlinear constitutive models in the large displacement analysis of beams". In: *Multibody system dynamics* 18.3 (2007), pp. 375–396.

[Mar+20] L Marx, MAF Gsell, A Rund, F Caforio, AJ Prassl, G Toth-Gayor, T Kuehne, CM Augustin, and G Plank. "Personalization of electro-mechanical models of the pressure-overloaded left ventricle: fitting of Windkessel-type afterload models". In: *Philos. Trans. A Math. Phys. Eng. Sci.* 378.2173 (2020), p. 20190342.

[Mir+16] GR Mirams, P Pathmanathan, RA Gray, P Challenor, and RH Clayton. "Uncertainty and variability in computational and mathematical models of cardiac physiology". In: *J. Physiol.* 594.23 (2016), pp. 6833–6847.

[Mol+18] R Molléro, X Pennec, H Delingette, A Garny, N Ayache, and M Sermesant. "Multifidelity-CMA: a multifidelity approach for efficient personalisation of 3D cardiac electromechanical models". In: *Biomechanics and modeling in mechanobiology* 17.1 (2018), pp. 285–300.

[NMA15] F Negri, A Manzoni, and D Amsallem. "Efficient model reduction of parametrized systems by matrix discrete empirical interpolation". In: *Journal of Computational Physics* 303 (2015), pp. 431–454.

[Nor+11] DA Nordsletten, SA Niederer, MP Nash, PJ Hunter, and NP Smith. "Coupling multi-physics models to cardiac mechanics". In: *Progress in biophysics and molecular biology* 104.1-3 (2011), pp. 77–88.

[PM] S Pagani and A Manzoni. "Enabling forward uncertainty quantification and sensitivity analysis in cardiac electrophysiology by reduced order modeling and machine learning". In: *Int. J. Numer. Methods Biomed. Engng* 37.6 (), e3450.

[PMQ18] S Pagani, A Manzoni, and A Quarteroni. "Numerical approximation of parametrized problems in cardiac electrophysiology by a local reduced basis method". In: *Computer Methods in Applied Mechanics and Engineering* 340 (2018), pp. 530–558.

[PW16] B Peherstorfer and K Willcox. "Data-driven operator inference for nonintrusive projection-based model reduction". In: *Computer Methods in Applied Mechanics and Engineering* 306 (2016), pp. 196–215.

[Pfa+20] MR Pfaller, M Cruz Varona, J Lang, C Bertoglio, and WA Wall. "Using parametric model order reduction for inverse analysis of large nonlinear cardiac simulations". In: *Int. J. Numer. Meth. Biomed. Engng.* 36.4 (2020), e3320.

[Pie+21] R Piersanti, PC Africa, M Fedele, C Vergara, L Dede', AF Corno, and A Quarteroni. "Modeling cardiac muscle fibers in ventricular and atrial electrophysiology simulations". In: *Computer Methods in Applied Mechanics and Engineering* 373 (2021), p. 113468.

[Pin85] A Pinkus. *n-Widths in Approximation Theory*. Berlin-Heidelberg: Springer-Verlag, 1985.

[Qia+19] E Qian, B Kramer, AN Marques, and KE Willcox. "Transform & Learn: A data-driven approach to nonlinear model reduction". In: *AIAA Aviation 2019 Forum*. 2019, p. 3707.

[Qua+19] A Quarteroni, L Dede', A Manzoni, and C Vergara. *Mathematical Modelling of the Human Cardiovascular System. Data, Numerical Approximation, Clinical Applications*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2019.

[QMN16] A Quarteroni, A Manzoni, and F Negri. *Reduced Basis Methods for Partial Differential Equations. An Introduction*. 2016.

[QMV17] A Quarteroni, A Manzoni, and C Vergara. "The cardiovascular system: mathematical modelling, numerical algorithms and clinical applications". In: *Acta Numerica* 26 (2017), pp. 365–590.

[QV08] A Quarteroni and A Valli. *Numerical approximation of partial differential equations*. Vol. 23. Springer Science & Business Media, 2008.

[RR16] A Radermacher and S Reese. "POD-based model reduction with empirical interpolation applied to nonlinear elasticity". In: *Int. J. Numer. Meth. Engng.* 107.6 (2016), pp. 477–495.

[RPK19] M Raissi, P Perdikaris, and GE Karniadakis. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". In: *Journal of Computational Physics* 378 (2019), pp. 686–707.

[RDQ20a] F Regazzoni, L Dede', and A Quarteroni. "Biophysically detailed mathematical models of multiscale cardiac active mechanics". In: *PLoS computational biology* 16.10 (2020), e1008294.

[RDQ20b] F Regazzoni, L Dede', and A Quarteroni. "Machine learning of multiscale active force generation models for the efficient simulation of cardiac electromechanics". In: *Computer Methods in Applied Mechanics and Engineering* 370 (2020), p. 113268.

[Reg+20a] F Regazzoni, M Salvador, PC Africa, M Fedele, L Dede', and A Quarteroni. "A cardiac electromechanics model coupled with a lumped parameters model for closed-loop blood circulation. Part I: model derivation". In: *arXiv preprint arXiv:2011.15040* (2020).

[Reg+20b] F Regazzoni, M Salvador, PC Africa, M Fedele, L Dede', and A Quarteroni. "A cardiac electromechanics model coupled with a lumped parameters model for closed-loop blood circulation. Part II: numerical approximation". In: *arXiv preprint arXiv:2011.15051* (2020).

[Reg+21] F Regazzoni, M Salvador, L Dede', and A Quarteroni. "A machine learning method for real-time numerical simulations of cardiac electromechanics". In: *arXiv preprint arXiv:2110.13212* (2021).

[RC04] CP Robert and G Casella. *Monte Carlo statistical methods*. Vol. 2. Springer, 2004.

[RCSR19] R Rodríguez-Cantano, J Sundnes, and ME Rognes. "Uncertainty in cardiac myofiber orientation and stiffnesses dominate the variability of left ventricle deformation response". In: *Int. J. Numer. Methods Biomed. Engng.* 35.5 (2019), e3178.

[Ros+14] S Rossi, T Lassila, R Ruiz-Baier, A Sequeira, and A Quarteroni. "Thermodynamically consistent orthotropic activation model capturing ventricular systolic wall thickening in cardiac electromechanics". In: *European Journal of Mechanics-A/Solids* 48 (2014), pp. 129–142.

[Sag88] K Sagawa. *Cardiac Contraction and the Pressure-volume Relationship*. Oxford University Press, USA, 1988.

[Sal02] A Saltelli. "Making best use of model evaluations to compute sensitivity indices". In: *Computer physics communications* 145.2 (2002), pp. 280–297.

[Sal+08] A Saltelli, M Ratto, T Andres, F Campolongo, J Cariboni, D Gatelli, M Saisana, and S Tarantola. *Global sensitivity analysis: the primer*. John Wiley & Sons, 2008.

[Sal+21] M Salvador, M Fedele, PC Africa, E Sung, A Prakosa, J Chrispin, NA Trayanova, and A Quarteroni. "Electromechanical modeling of human ventricles with ischemic cardiomyopathy: numerical simulations in sinus rhythm and under arrhythmia". In: *Computers in Biology and Medicine* 136 (2021), p. 104674.

[Sat+98] T Sato, T Shishido, T Kawada, H Miyano, H Miyashita, M Inagaki, M Sugimachi, and K Sunagawa. "ESPVR of in situ rat left ventricle shows contractility-dependent curvilinearity". In: *American Journal of Physiology-Heart and Circulatory Physiology* 274.5 (1998), H1429–H1434.

[Sch89] E Schmidt. "Zur Theorie der linearen und nichtlinearen Integralgleichungen". In: *Integralgleichungen und Gleichungen mit unendlich vielen Unbekannten*. Springer, 1989, pp. 190–233.

[Sho15] RM Shoucri. “End-systolic Pressure–Volume Relation, Ejection Fraction, and Heart Failure: Theoretical Aspect and Clinical Applications”. In: *Clinical Medicine Insights: Cardiology* 9 (2015), CMC–S18740.

[Smi13] RC Smith. *Uncertainty quantification: theory, implementation, and applications*. Vol. 12. Siam, 2013.

[Sob90] IM Sobol’. “On sensitivity estimation for nonlinear mathematical models”. In: *Matematicheskoe modelirovanie* 2.1 (1990), pp. 112–118.

[Sob67] IM Sobol’. “On the distribution of points in a cube and the approximate evaluation of integrals”. In: *Zhurnal Vychislitel’noi Matematiki i Matematicheskoi Fiziki* 7.4 (1967), pp. 784–802.

[SJ+69] DD Streeter Jr, HM Spotnitz, DP Patel, J Ross Jr, and EH Sonnenblick. “Fiber orientation in the canine left ventricle during diastole and systole”. In: *Circulation research* 24.3 (1969), pp. 339–347.

[Sul15] TJ Sullivan. *Introduction to uncertainty quantification*. Vol. 63. Springer, 2015.

[Swi+19] R Swischuk, L Mainini, B Peherstorfer, and K Willcox. “Projection-based model reduction: Formulations for physics-based machine learning”. In: *Computers & Fluids* 179 (2019), pp. 704–717.

[TR13] P Tiso and DJ Rixen. “Discrete empirical interpolation method for finite element structural dynamics”. In: *Topics in Nonlinear Dynamics, Volume 1*. Springer, 2013, pp. 203–212.

[TD18] GJ Tortora and BH Derrickson. *Principles of anatomy and physiology*. John Wiley & Sons, 2018.

[Tra11] NA Trayanova. “Whole-heart modeling: applications to cardiac electrophysiology and electromechanics”. In: *Circulation research* 108.1 (2011), pp. 113–128.

[WFC85] L K Waldman, YC Fung, and JW Covell. “Transmural myocardial deformation in the canine left ventricle. Normal in vivo three-dimensional finite strains.” In: *Circulation research* 57.1 (1985), pp. 152–163.

[Wan+13] HM Wang, H Gao, XY Luo, C Berry, BE Griffith, RW Ogden, and TJ Wang. “Structure-based finite strain modelling of the human left ventricle in diastole”. In: *Int. J. Numer. Meth. Biomed. Engng.* 29.1 (2013), pp. 83–103.

[WLW09] N Westerhof, JW Lankhaar, and BE Westerhof. “The arterial windkessel”. In: *Medical & biological engineering & computing* 47.2 (2009), pp. 131–141.

[Wig51] CJ Wiggers. “Physiology from 1900 to 1920: incidents, accidents, and advances”. In: *Annual review of physiology* 13.1 (1951), pp. 1–21.

[WSH14] D Wirtz, DC Sorensen, and B Haasdonk. “A posteriori error estimation for DEIM reduced nonlinear dynamical systems”. In: *SIAM Journal on Scientific Computing* 36.2 (2014), A311–A338.

[WK14] J Wong and E Kuhl. “Generating fibre orientation maps in human heart models using Poisson interpolation”. In: *Computer methods in biomechanics and biomedical engineering* 17.11 (2014), pp. 1217–1226.

[Zin+21] A Zingaro, I Fumagalli, L Dede, M Fedele, PC Africa, AF Corno, and A Quarteroni. "A multiscale CFD model of blood flow in the human left heart coupled with a lumped-parameter model of the cardiovascular system". In: *arXiv preprint arXiv:2110.02114* (2021).